
Gouvernance des projets open source : le cas du logiciel Claroline

Robert Viseur¹, Amel Charleux²

1. Université de Mons

Faculté Warocqué d'Economie et de Gestion
17, Place Warocqué, 7000 Mons, Belgique
robert.viseur@umons.ac.be

2. Université de Montpellier

Montpellier Recherche en Management (MRM)
Espace Richter (Bâtiment B), Rue Vendémiaire, CS19519, 34960 Montpellier
amel.charleux@umontpellier.fr

RESUME. Claroline est un projet de Learning Management System open source initié en Belgique par l'Université Catholique de Louvain. Plusieurs projets open source en sont, directement ou indirectement, dérivés (Dokeos, Chamilo, Claroline Connect). Compte tenu de sa diffusion, des opportunités de réalisation d'une étude longitudinale complète, de l'évolution de sa gouvernance, de ses forks et de sa résilience, Claroline présente un terrain de recherche idéal pour comprendre les dynamiques communautaires dans les communautés open source ainsi que les modalités de cohabitation avec des éditeurs publics ou privés. Dans cet article, nous proposons les résultats préliminaires d'une étude de cas basée sur des entretiens semi-directifs portant sur les modalités de gouvernance et de changement de modèle d'affaires au sein d'un écosystème open source. Nous montrons en particulier comment les choix successifs de gouvernance peuvent conduire à des mouvements de reconfiguration des communautés.

ABSTRACT. Claroline is an open source Learning Management System project initiated in Belgium by the Catholic University of Louvain. Several open source projects are, directly or indirectly, derived (Dokeos, Chamilo, Claroline Connect). Through its dissemination, the opportunities for conducting a comprehensive longitudinal study, the evolution of its governance, its forks and its resilience, Claroline presents an ideal research ground for understanding the community dynamics in open source communities as well as the modalities of cohabitation with public or private organizations. In this article, we propose the preliminary results of a case study based on semi-structured interviews about governance and business model changes in an open source ecosystem. In particular, we show how successive choices of governance can lead to community reconfiguration movements.

MOTS-CLES : open source, gouvernance, communauté, fork, innovation.

KEYWORDS: open source, governance, community, fork, innovation.

1. Introduction

Les logiciels libres et open source ont connu une forte extension dès le milieu des années 90 avec l'essor du web, le développement des grandes communautés (p.ex. Apache) et l'implication croissante des entreprises (p.ex. Netscape ou IBM). Ils constituent par ailleurs un objet d'étude depuis près de 20 ans. Les recherches ont ainsi conduit à une bonne compréhension des modèles d'affaires open source (p.ex. dual licensing) et de leurs évolutions (p.ex. cloud computing et tendance « as a service »). La compréhension du fonctionnement des communautés open source reste par contre partielle malgré l'engouement récent pour les approches quantitatives (p.ex. data mining de répertoires de codes sources). Les conditions de la cohabitation entre les communautés et les éditeurs (qu'il s'agisse d'entités commerciales ou non commerciales) restent également mal connues, soient que les problèmes soulevés ne trouvent pas d'échos dans la littérature scientifique, soient que la traduction des solutions proposées s'avère difficile.

L'étude de cas porte sur le logiciel Claroline ainsi que sur les logiciels qui en ont été dérivés (Dokeos, Chamilo, Claroline Connect). Claroline est un logiciel open source de type Learning Management System (LMS) développé à partir de l'an 2000 au sein de l'Université Catholique de Louvain (Belgique), très vite rejointe par l'ECAM, l'Institut Supérieur Industriel associé à la HE Léonard de Vinci, puis repris par le Consortium Claroline à partir de 2007. Cet écosystème logiciel s'avère intéressant à plus d'un titre, en particulier : (1) l'accès aisé au terrain et l'opportunité de réalisation d'une étude longitudinale couvrant la vie du projet (depuis sa création), (2) la transformation progressive de la gouvernance du projet initial et (3) la survenance de plusieurs forks et l'éclatement de la communauté en plusieurs sous-communautés concurrentes.

Nous présentons dans cet article les résultats préliminaires de cette étude de cas. Nous chercherons en particulier à répondre à la question suivante : *“Comment les communautés open source réagissent-elles aux choix et aux changements de gouvernance ?”*.

Notre article est décomposé en cinq sections. Dans une première section, nous procéderons à un état de l'art sur la gouvernance des projets open source, leurs facteurs de succès ainsi que sur les modèles d'affaires. Dans une seconde, nous présenterons la méthodologie utilisée pour l'étude de cas. Dans une troisième section, nous présentons l'étude de cas proprement dites ainsi que ses résultats. Nous les complétons ensuite par les modalités d'organisation et les résultats d'un atelier créatif. Dans une quatrième section, nous discutons les résultats. Dans une cinquième et dernière section, nous concluons par une synthèse des résultats et par les futures perspectives de recherche.

3. Etat de l'art

L'open source est aujourd'hui une réalité quotidienne dans le secteur informatique. Selon une étude réalisée en 2015 pour le Conseil National du Logiciel

Libre, le secteur du logiciel libre et open source représenterait ainsi en France un chiffre d'affaires cumulé de 4,1 milliards d'euros (2015) pour environ 50000 emplois (CNLL, 2015). Parmi les avantages cités par la centaine d'entreprises ayant participé à l'étude, citons un *"facteur d'innovation sans équivalent avec les logiciels propriétaires"* loué par 75% des répondants.

L'investissement croissant des entreprises dans les projets open source a notamment été analysée par Fitzgerald (2006). Quant à la question de l'existence ou non de modèles d'affaires open source, elle a été posée voici quelques années (Vasquez Bronfman et Miralles, 2007). Il ressortait de cette interrogation que la majorité des prestataires démarraient leur activité en tant que fournisseurs de services et sans réelle réflexion préalable sur le modèle d'affaires. Ce constat n'a rien de surprenant. Teece (2010) estime ainsi que *"le bon modèle d'affaires est rarement apparent tout de suite dans les industries émergentes"*. Le modèle d'affaires décrit les modalités de création, de distribution et de capture de valeur. Il peut être générique, partagé par plusieurs entreprises parfois en concurrence (Teece, 2010). Dans le cas open source, certains modèles d'affaires plus originaux se distinguent, comme le principe de la double licence (dual licensing) étudié par Välimäki (2003). Viseur (2013a) présente une synthèse de ces modèles d'affaires et met en évidence le caractère non figé de ces modèles, pouvant conduire à des conflits avec la communauté rassemblée autour du logiciel, par exemple sous la forme d'un refus de changement de licence (Viseur et Robles, 2015). Ils évoluent ainsi avec la maturité de l'entreprise (p.ex. transformation vers un rôle d'éditeurs et structuration d'un réseau d'intégrateurs) ou des évolutions technologiques (p.ex. popularisation du cloud computing et succès des offres de type Software as a Service).

Les communautés open source sont susceptibles d'attirer des acteurs fort différents : passionnés, informaticiens indépendants, informaticiens salariés (secteurs privé ou public), chercheurs,... Lakhani et al. (2002) ont très tôt clarifié les motivations individuelles (contributeurs). Ils ont pu constater que les développeurs étaient souvent attirés vers l'open source par pragmatisme (top 3 : stimulation intellectuelle, amélioration de l'expertise et fonctionnalités), estimant pour un tiers d'entre eux que le code doit être open source, attendant des responsables capables de dialoguer, de fournir une vision et de produire ou intégrer du code source. Si les contributeurs ne sont pas des "croisés" (battre Microsoft apparaissait comme une motivation pour un peu plus d'un contributeur sur dix seulement), ils sont donc attachés à certaines valeurs garanties par les licences des logiciels open source. L'entreprise (p.ex. éditeur) doit donc composer avec cette culture ou faire le choix d'attirer des contributeurs pour lesquels la liberté du code est moins importante (p.ex. développeurs rémunérés). La participation de contributeurs rémunérés permettrait par ailleurs d'accroître l'activité sur les projets sans dégrader la qualité du code source (Roberts et al., 2006). L'open source apparaît dès lors comme un environnement complexe réconciliant motivations intrinsèques et extrinsèques. Les motivations des entreprises peuvent être de natures sociales, économiques ou technologiques, avec une prédominance des deux dernières (Bonaccorsi et Rossi, 2006 ; Dahlander et Magnusson, 2005 ; Feller et Fitzgerald, 2002). Il s'agit en particulier des opportunités d'innovation pour les petites entreprises, de la collecte de

feedback (p.ex. rapports d'erreurs), de la fiabilité et la qualité des logiciels open source et de la plus grande indépendance vis-à-vis des grandes entreprises (p.ex. prix et licences). La conformité aux valeurs du mouvement du logiciel libre arrive en cinquième position. Au delà des tensions possibles autour de l'interprétation des normes sociales et des valeurs internes aux communautés, il semble donc exister une adhésion aux logiques d'ouverture et de collaboration propres aux logiciels libres et open source, tant chez les développeurs qu'au sein des entreprises.

Les modalités d'interactions entre l'organisation assurant l'édition du logiciel (p.ex. éditeur ou fondation) et la communauté qui en soutient le développement varient fortement d'un projet à l'autre (p.ex. MySQL ou Eclipse) et ne sont pas figées dans le temps (p.ex. Netscape / Mozilla). La gouvernance du projet, soit les conditions d'exercice du pouvoir au sein de la communauté ou de l'écosystème, fait l'objet d'études depuis une dizaine d'années sur le plan de la structuration au cours du temps (De Laat, 2007) ou des dimensions de cette structuration (Markus, 2007). Sur cette base, Viseur (2016) propose quatre logiques de gouvernance open source : (1) la logique informelle (cas général des petits projets hébergés sur des dépôts publics sans règles formelles autres que la licence), (2) la logique commerciale (cas des éditeurs open source souhaitant conserver le contrôle du projet), (3) la logique communautaire (cas des projets de grande taille ou critiques pérennisés par une entité légale) et (4) la logique industrielle (cas des acteurs industriels poursuivant un objectif de mutualisation dans un cadre coopératif).

Les tensions entre le responsable d'un projet (p.ex. éditeur) et la communauté qui le soutient peut conduire à une scission de la communauté. Pour les projets open source, ce processus est baptisé "fork". Au travers de l'analyse de 26 forks populaires, Viseur (2012, 2016) a étudié les conséquences d'un fork sur le projet initial ainsi que les motivations qui y conduisent. La cohabitation entre les deux projets, c'est-à-dire le projet original et son fork, représente la situation la plus fréquente. Les motivations apparaissent par contre très diverses : arrêt du développement (sous sa forme open source), objectifs techniques divergents, changement de licence, conflit autour de l'utilisation d'une marque, conflit autour de la gouvernance du projet (capacité d'influencer le projet au travers de sa feuille de route ou de ses contributions), les différences "culturelles" (hétérogénéité de la communauté) et la recherche d'innovation. Les motivations liées à la capacité à influencer le projet et aux objectifs techniques interviennent dans plus des trois quarts des forks étudiés. Sur le plan théorique, les forks dus à des divergences d'objectifs techniques ou fonctionnels pourraient être évités par le développement d'une architecture modulaire permettant la personnalisation ou la verticalisation sous la forme de distributions (Viseur, 2016). MacCormack et al. (2006) parlent d'architecture de participation. L'impact des logiques de gouvernance sur le risque de fork reste par contre incertain.

3. Méthodologie

L'étude de cas de l'écosystème du logiciel open source Claroline nous a conduit à travailler sur 4 logiciels distincts : Claroline (2001) et son héritier direct Claroline

Connect (2014), le fork Dokeos (2003) et son fork Chamilo (2010). Elle s'apparente à un cas unique et longitudinal (Thietart, 2007 ; Yin, 2009).

Dans une première phase, huit entretiens semi-directifs, généralement individuels, en face-à-face, ont été réalisés, en binôme, sur base d'un guide d'entretien standard, avec prise de note et enregistrement. Les personnes interviewées couvrent les responsables actuels des différents projets ainsi que les principaux initiateurs du projet Claroline. De cette manière, la contamination intragroupe a été limitée par l'accès à des sous-groupes suffisamment disjoints car issus du projet initial et de ses scissions successives (forks). L'arrêt des interviews a été dicté par l'atteinte d'une saturation en informations. L'entretien démarrait par une question générale permettant de rester non-directif. Des questions plus précises étaient ensuite utilisées pour relancer les échanges ou préciser un point. Une attention particulière a été accordée à la liberté de parole des interviewés, en particulier lorsque des divergences de vue apparaissaient (p.ex. conflits). Les entretiens doivent faire l'objet d'une retranscription et d'un codage. Ce travail n'a cependant pas été réalisé, les résultats préliminaires, validés par les chercheurs (regards croisés), se basant essentiellement sur les notes d'entretiens et les enregistrements (vérification).

Outre les entretiens, des sources primaires et secondaires ont été utilisées, en particulier pour établir la chronologie des différents projets (p.ex. site web des projets, présentations publiques et rapports professionnels). Le projet Claroline a fait l'objet d'une première étude de cas par questionnaire en 2006 (Viseur, 2007). La totalité du matériel de recherche (questionnaire, réponses au questionnaire, références, sitographie, notes, enregistrements, résultats préliminaires,...) fait l'objet d'un partage entre les chercheurs.

Dans une seconde phase, une première présentation de l'historique, des résultats préliminaires de l'étude ainsi que des questions soulevées a été assurée par les auteurs de l'étude à l'occasion de la conférence annuelle réunissant à Bruxelles les utilisateurs du logiciel (ACCU 2017). A cette occasion, un atelier créatif a également été réalisé. Baptisé "*Recharge ton ACCU*", il visait à identifier des pistes de redynamisation de la communauté gravitant autour de Claroline Connect. Cette phase permettait par ailleurs un élargissement de l'accès au terrain et la captation du point de vue des utilisateurs proches du projet. La posture adoptée s'apparente à une posture constructiviste transformative, où les savoirs construits par les chercheurs sont confrontés à la perception des acteurs de terrain (Giordano, 2003).

4. Etudes de cas

4.1. Historique des projets

L'histoire du projet Claroline / Claroline Connect peut être découpée en quatre grandes étapes.

Tableau 1. Historique du projet Claroline

Année	Evènement marquant
2000	Démarrage des développements qui donneront naissance à Claroline au sein de l'IPM (UCL).
2001	Publication de la première version de Claroline.
2003	Départ de Thomas De Praetere pour former le fork Dokeos et la société Dokeos.
2004	Obtention d'un financement public (projet WIST) par l'ECAM.
2007	Création du consortium Claroline (fondation).
2010	Fork Chamilo issu de Dokeos.
2010	Sortie de la note d'orientation de Claroline Connect.
2012	Premiers développements de Claroline Connect.
2014	Sortie de Claroline Connect (RC).
2015	Création de la société Formalibre.

L'ère des pionniers (2000 - 2003)

En l'an 2000, l'Université Catholique de Louvain (UCL) cherche à mettre en place des services en ligne dédiés à l'elearning. Le logiciel propriétaire WebCT est retenu et l'Institut de Pédagogie universitaire et des Multimédias (IPM), devenu Louvain Learning Lab (LLL) en 2015, chargé de l'accompagnement des enseignants. Cette solution se révèle coûteuse, lente et peu adaptée aux besoins. Une petite équipe (2 développeurs) se lance dans le développement d'une solution, basée sur des composants open source, centrée sur les usages. Cette solution, bientôt baptisée Claroline, est publiée en 2001. Elle rencontre un vif succès tant à l'intérieur qu'à l'extérieur de l'institution. Sur le plan théorique, ces pionniers peuvent être assimilés à des utilisateurs de pointe décidant d'innover par eux-mêmes pour disposer d'une solution adaptée à leur besoin (Franke & Von Hippel, 2003). Cette période se termine avec le départ de Thomas de Praetere, un des initiateurs du projet, caractérisé par un profil d'intrapreneur, désireux de créer une entreprise de services autour du logiciel Claroline. En conflit avec la structure en charge de l'accompagnement des spin-offs, il décide de forker le projet, le rebaptise et lance la société Dokeos.

La pérennisation et l'institutionnalisation (2004 - 2007)

L'énergie des pionniers laisse la place à une consolidation et une recherche de ses conditions de pérennisation (p.ex. financement). Cela se traduit, d'une part, par l'arrivée d'un nouveau partenaire, l'ECAM (HE Vinci), et le dépôt d'un projet WIST permettant le financement du projet, et, d'autre part, le renfort du projet par une équipe de professionnels de l'informatique. De manière à maintenir le contrôle de la feuille de route (roadmap) et la qualité des développements, la priorité est

donnée à des pôles de développeurs issus des institutions partenaires plutôt qu'à l'activité communautaire. Cette période prend fin avec l'arrivée à terme du WIST et les difficultés à trouver de nouveaux financements.

L'autonomisation et la quête d'une gouvernance (2008 - 2011)

Le projet s'autonomise progressivement des institutions qui l'ont créées. Formellement créé en 2007, le consortium Claroline fournit un cadre permettant de veiller aux intérêts du projet Claroline. Le logiciel Claroline est stabilisé. Les lignes directrices de sa nouvelle version, Claroline Connect, sont présentées en 2010. En parallèle, le projet Dokeos ferme progressivement son modèle d'affaires, évoluant vers un modèle de double licence peu ouvert à la communauté ; son fork Chamilo voit le jour en 2010 et se structure en une association ouverte à la communauté et aux contributeurs internationaux.

Le redéploiement local et international (depuis 2012)

A partir de 2012, le logiciel Claroline Connect est mis en développement. En 2014, le projet doit faire face à l'arrêt du soutien de l'Université Catholique de Louvain (UCL) et à sa migration vers le logiciel open source Moodle. Cette défection est compensée par l'arrivée de l'Université de Lyon I, via le service iCAP (Innovation Conception Accompagnement pour la Pédagogie), qui apporte l'expertise et les ressources associées au logiciel Spiral Connect¹. En 2014 également, les premières versions utilisables de Claroline Connect sont présentées, avec une première version stable officiellement lancée fin mai 2015. A cette occasion, un prestataire privé, baptisé Formalibre, est également créé. Le projet Claroline Connect peut se redéployer et, en particulier, relancer sa communication vers les utilisateurs.

4.2. Gouvernance et forks

Le projet Claroline et ses dérivés présentent des gouvernances distinctes et reflètent les logiques de gouvernance identifiées par Viseur (2016).

Logique informelle

A sa naissance, Claroline suit une logique informelle. L'équipe de développement est autonome. Les règles sont définies par la licence (GPL).

Logique commerciale

L'éditeur du fork Dokeos évolue rapidement vers une logique commerciale. La communauté est perçue comme un frein en termes de rentabilité et de time-to-market. Le projet se referme progressivement jusqu'à proposer un modèle de double licence (sans réelle communauté associée au développement) puis un re-développement sous licence propriétaire à la suite du fork Chamilo.

Logique industrielle

¹ Le nom Claroline Connect est une contraction de Claroline et Spiral Connect.

Le fonctionnement du consortium Claroline confirme la préférence pour les pôles de développeurs, apportés par des institutions partenaires. L'accès au statut de Membre est ainsi conditionné à l'apport de ressources (p.ex. développeurs ou financements) sur le projet. Ce choix conduit, sans intention maligne, à écarter certaines franges de la communauté.

Logique communautaire

Le projet Chamilo récupère dès son fork une partie de la communauté d'utilisateurs et mise d'emblée sur une diffusion maximale du projet (p.ex. installation en 1 clic). Si le fonctionnement de l'association peut sembler similaire à celui du consortium Claroline, elle se révèle cependant ouverte aux membres actifs (méritocratie) sans condition d'apport de ressources.

Tableau 2. Claroline, la gouvernance et les forks

Gouvernance	Forks	Editeurs
Logique informelle	----- Claroline (2001)	
Logique commerciale	 Dokeos (2003)	Dokeos
Logique communautaire	 Chamilo (2010)	Beeznest, ...
Logique industrielle	 Claroline Connect (2014)	Formalibre (2015), ...

Forks (Dokeos et Chamilo)

L'écosystème Claroline a été traversé par différents soubresauts. Le premier est provoqué par le fork Dokeos, issu de Claroline. Le second est Chamilo, issu de Dokeos. Il n'y a pas d'échanges de code entre variantes, malgré des tentatives de rapprochement entre les équipes de Dokeos, de Claroline Connect et de Chamilo.

Le fork Dokeos a été initié par Thomas de Praetere, un des créateurs du projet Claroline. Désireux de lancer une entreprise de services, il entre en conflit avec la structure en charge de l'accompagnement des spin-offs au sein de l'université à propos des conditions de valorisation de la marque Claroline (condition standard de prise de participation dans la spin-off contre un droit d'utilisation de la marque). Le fork Dokeos est accolé à l'entreprise éponyme.

Le fork Chamilo a été initié par Yannick Warnier, un temps partenaire de la société Dokeos en Amérique latine et fondateur de la société Beeznest. Ce fork est motivé par la fermeture progressive du projet Dokeos ; ses initiateurs souhaitent une

ouverture accrue vis-à-vis de la communauté. La création d'une association est voulue comme un moyen de garantir cette ouverture sur le long terme et de limiter le pouvoir des initiateurs du projet Chamilo.

4.3. Résistances au changement

Des résistances à différents changements ont par ailleurs été constatées.

1) Innover avec les utilisateurs (Claroline)

Les initiateurs du projet Claroline sont des utilisateurs de pointe confrontés à la politique de l'institution (sélection du logiciel propriétaire WebCT jugé insatisfaisant). Portés par les utilisateurs (enseignants), favorables à l'innovation centrée sur les utilisateurs plutôt qu'à une approche fonctionnelle, ils s'opposent à la structure soucieuse de faire respecter ses choix technologiques. Un parallèle pourrait être dressé avec le projet belge CommunesPlone, porté par des informaticiens communaux contre des choix ministériels.

2) Pérenniser le projet (Claroline)

Le projet se structure en vue d'assurer sa pérennisation. Les pionniers se sentent dépossédés du projet qu'ils ont créés. Il en résulte le fork du projet par un des fondateurs² puis le départ du second développeur.

3) Privatiser le projet (Dokeos)

Dokeos fait évoluer son modèle d'affaires. L'entreprise tente une privatisation partielle du projet par le passage à un modèle de double licence et l'ajout de modules propriétaires permettant de différencier les versions communautaires et commerciales. Il en résulte un second fork, baptisé Chamilo, rassemblant la communauté.

4) Fusionner deux variantes (Claroline Connect et Chamilo)

Le fork Chamilo et la validation de la feuille de route de Claroline Connect sont contemporains. Une fusion de Chamilo et de Claroline est dès lors tentée. Il s'agit d'un échec, dont les raisons mériteraient un approfondissement. Les différences d'approche qualité et d'architecture font partie des motifs (path dependency ?). Un parallèle pourrait être dressé avec Nokia et la fusion, tardivement réussie, de Maemo et Moblin au sein de Meemo, aujourd'hui devenu Sailfish OS, marquée par les difficultés d'homogénéisation des pratiques entre partenaires et communautés (Viseur, 2013b).

5) Rassembler la communauté (Claroline Connect)

² Dans un projet open source, la propriété prend différentes formes. Le partage du code source est régulé par la licence mais peut cohabiter avec la pleine propriété d'une marque. La nature open source permet ici à un employé de s'affranchir de son employeur suite à un désaccord sur les conditions d'exploitation du logiciel.

Le basculement sur la nouvelle version de Claroline, baptisée Claroline Connect, apportant différentes évolutions notamment en matière d'innovation pédagogique, s'est accompagné de ruptures radicales chez les institutions utilisatrices, sous la forme notamment de migrations vers le logiciel open source Moodle (p.ex. UCL). Les causes des résistances au changement à la migration vers Claroline Connect sont multiples (mauvaise communication, rapports de force internes aux organisations, temps d'attente de la nouvelle version,...).

6) Aligner les stratégies (Claroline Connect)

La création de Formalibre, du fait de sa capacité de production et de ses sources de financement (opérateurs privés), pourrait entraîner de nouvelles formes de résistances dues aux objectifs divergents entre l'entreprise privée (p.ex. priorité aux clients privés) et les institutions membres ou utilisatrices (p.ex. complexité d'installation de la solution actuelle et objectif d'innovation pédagogique).

4.4. Relations à la communauté

Ces résultats préliminaires apportent de nouveaux éléments, incluant de nouvelles interrogations, principalement relatives à trois thématiques : la gouvernance, la communication et l'animation de la communauté.

Gouvernance

La création du consortium Claroline (fondation internationale) a apporté plusieurs bénéfices pour le projet. Premièrement, le consortium permet la négociation des règles. Le fonctionnement du projet ne se fait plus en fonction d'un leader plus ou moins éclairé mais suivant des règles discutées et amendées suivant des procédures prédéfinies. Deuxièmement, le consortium permet de centraliser la gestion des ressources. Il permet la collecte et la mise à disposition des financements dans un sens décidé collectivement au travers du conseil d'administration (CA) et de son assemblée générale (AG) annuelle. Troisièmement, le consortium garantit la neutralité du projet. La fondation ne se confond pas avec ses membres. L'étiquette "catholique" associée à l'UCL peut ainsi amener des difficultés inattendues, que ce soit en local (p.ex. réminiscence des guerres scolaires en Belgique) ou à l'international (p.ex. états confessionnels non chrétiens). Quatrièmement, le consortium apporte davantage de stabilité au projet. La fondation garantit une pérennité face à des changements internes aux institutions membres ainsi qu'aux départs de personnalités importantes. De la sorte, elle réduit le risque perçu par les utilisateurs, notamment institutionnels. Cinquièmement, le consortium renforce l'attractivité du projet. L'existence d'une fondation, par laquelle le projet ne se confond pas sur le plan juridique avec une institution influente, rassure les partenaires potentiels qui hésitent ainsi moins à franchir le pas et à adopter la solution.

Communication

Le cas de Claroline / Claroline Connect, mais aussi d'autres projets open source étudiés, offrent le constat d'un manque d'efficacité de la communication autour du projet. Cela se traduit par la difficulté de communiquer de manière efficace vers les

différentes parties prenantes du projet : membres du consortium, utilisateurs institutionnels, développeurs tiers,... Dans le cas particulier de Claroline / Claroline Connect, nous posons le constat d'une connaissance très imparfaite de l'histoire du projet (et de ses forks) au sein des institutions d'enseignement (supérieur ou universitaire) mais aussi d'un manque de communication efficace dans le cadre de l'abandon du support de Claroline et de la migration vers Claroline Connect, conduisant à considérer la solution morte et enterrée. Ce dernier constat pourrait s'expliquer par les faibles ressources disponibles pour gérer la communication du projet mais aussi par la difficulté à diffuser le bon message au bon moment vers les très nombreuses parties-prenantes.

Animation

En matière d'animation, les interviews ont conduit à des interrogations sur les causes d'un manque de contributions externes sur le projet (excepté les nécessaires traductions). Est-ce dû à la priorité accordée dès le départ aux pôles de développeurs (collaboration entre institutions) plutôt qu'aux communautés de développeurs ? Est-ce dû à un manque d'animation quant à la possibilité de créer des extensions pour Claroline / Claroline Connect ? Est-ce dû à un manque d'animation du réseau de développeurs ou de prestataires de services développant sur le projet sans être membre du consortium ? Est-ce une stratégie délibérée qui permet de garder le contrôle sur les évolutions de la solution (roadmap) ?

4.5. Atelier "Recharge ton ACCU"

La participation à la conférence annuelle (ACCU 2017) a permis de confronter la compréhension de la situation suite aux entretiens à la perception de la communauté dans son ensemble, incluant des éléments jusqu'alors inconnus (p.ex. écoles secondaires). Les réactions ont notamment confirmé la préférence marquée, dès avant la création du consortium, pour les pôles de développeurs, conduisant à une absence de dynamisation de la communauté des développeurs.

A la suite de ces discussions, un atelier a été organisé avec une dizaine de participants selon le processus suivant. Dans un premier temps, un brainstorming a été réalisé sur base de la question suivante : "*Comment améliorer la vitalité de la communauté Claroline Connect ?*". Les participants, au nombre de dix environ, stimulés par des inducteurs visuels (mots projetés), étaient invités à proposer des améliorations en termes de communication et d'animation de la communauté. Cette séance a débouché sur quarante idées environ, regroupées en 12 propositions. Dans un second temps, ces 12 propositions ont été soumises à un vote secret (3 votes "pour", 3 votes "contre" et 1 "coup de coeur" par personne). Le caractère secret du vote permet d'éviter les effets de mimétisme et de plus facilement faire apparaître les divergences. Les idées polémiques peuvent ensuite être discutées.

Cette séance a permis l'émergence de deux possibles actions prioritaires : (1) la communication sur les éléments de différenciation de Claroline Connect et (2) le développement de la documentation et de l'entraide en ligne (p.ex. forums). Elle a également permis la mise en évidence de blocages existants en matière de représentation de certaines franges de la communauté (p.ex. utilisateurs issus de

l'enseignement secondaire), pouvant faire l'objet d'une analyse plus approfondie. En particulier, le manque de représentation (memberships) des contributeurs hors institutions membres du consortium pourrait expliquer le manque de participation (West et O'Mahony, 2008) ; quant au manque d'animation et d'attention à la diffusion du logiciel (p.ex. effectivité des procédures d'installation et promotion des outils de migration), il réduit tant les bénéfices directs (p.ex. contributions en code) qu'indirects (p.ex. augmentation du nombre d'utilisateurs).

5. Discussion

Ces résultats préliminaires apportent aussi des réponses quant à la question des réactions de la communauté aux choix et aux changements de gouvernance en lien avec l'évolution des modèles d'affaires.

Modèles d'affaires et logiques de gouvernance

S'ils présentent de fortes similarités sur le plan fonctionnel, Claroline et ses variantes présentent des spécificités sur le plan des modèles d'affaires.

Tableau 3. Modèles d'affaires associés aux projets

	Claroline	Dokeos	Chamilo	Claroline Connect
Capture de la valeur	Financements (partenaire et secteur public)	Projets commerciaux, double licence	Activités de services	Financements (partenaires), projets commerciaux (Formalibre)
Création de la valeur	Pôles de développeurs	Editeur	Mutualisation	Pôles de développeurs (incluant un éditeur)
Distribution de la valeur	Code source publié	Accès progressivement restreint	Accès au code source et installation simplifiée (p.ex. 1-clic)	Code source publié
Logique de gouvernance	Informelle, puis industrielle	Commerciale	Communautaire	Industrielle

Configuration des communautés

En cas de logique informelle, la communauté ne dispose pas de garde-fous culturels ou réglementaires, excepté la licence du logiciel, qui fixe des droits et des devoirs fondamentaux. Cette configuration nous paraît propice à une exacerbation des conflits, conduisant à la survenance de forks (p.ex. Dokeos). En cas de logique communautaire, la gouvernance veille à réguler, dans un souci de recherche d'équilibre, les divergences d'intérêts pouvant survenir entre membres de la communauté. Cette configuration nous paraît propice à un large rassemblement

d'utilisateurs et de développeurs (p.ex. Chamilo). Par contre, elle est plus difficilement conciliable avec les priorités d'organisations ayant une feuille de route nécessitant un contrôle minimum du projet (p.ex. contrainte forte de type time-to-market). Ce type d'organisation privilégiera (et s'orientera donc progressivement vers) une logique commerciale (p.ex. Dokeos) ou une logique industrielle (p.ex. Claroline). La conséquence d'un passage vers une logique commerciale ou industrielle est qu'elle tend à opérer un choix de segmentation (Table 3) parmi les partenaires et contributeurs, susceptible de conduire à un fork (p.ex. Chamilo) ou à une séparation progressive (p.ex. Claroline post-consortium). L'architecture du logiciel (p.ex. modularité) et l'animation de la communauté pourraient limiter cet effet d'éviction (p.ex. stimuler la création de modules et verticaliser sous la forme de distributions).

Tableau 4. Logique de gouvernance et configuration de la communauté

	Logique commerciale	Logique industrielle	Logique communautaire
Acteur(s) dominant(s)	Editeur	Grande(s) organisation(s)	Méritocratie égalitaire
Communauté	Partenaires (réseau structuré)	Grandes organisations (coopétition)	Petites organisations et utilisateurs individuels
Développeur(s) dominant(s)	Editeur et partenaires	Grandes organisations	Core team et développeurs individuels
Motivations	Rentabilité et time-to-market	Contrôle des développements	Mutualisation la plus large possible
Dangers	Fermeture progressive (p.ex. open core)	Déséquilibres entre partenaires	Accroissement des coûts de négociation

Formes de résistance au changement

En pratique, la communauté peut s'opposer à une situation ou à un changement de manière graduelle (1) en exprimant son mécontentement (p.ex. forums ou conférences), (2) en continuant à utiliser le logiciel mais sans plus y contribuer, (3) en cessant d'utiliser le logiciel (p.ex. migration) et (4) en organisant une scission de la communauté (fork).

Conclusion

Résumé

Claroline -et les projets qui en sont dérivés : Dokeos, Chamilo, Claroline Connect- fournit un terrain d'étude idéal pour la compréhension des mécanismes de

gouvernance et de transformation organisationnelle des projets open source. Cette recherche a permis de dresser l'historique des différents projets, de mieux comprendre les mécanismes conduisant aux forks, d'analyser les bénéfices associés à la création d'un consortium, d'identifier les difficultés associées à l'animation d'une communauté et, enfin, d'explorer les liens existant entre modèles d'affaires, logiques de gouvernance et configuration des communautés.

Transformation organisationnelle

Le projet Claroline illustre la difficulté de faire évoluer le projet tout en maintenant la cohésion de la communauté et en évitant les effets d'éviction. La pérennisation du projet implique des choix de modèle d'affaires et de gouvernance susceptibles d'éloigner certaines franges de la communauté. La prédilection pour les pôles de développeurs, pour des raisons de contrôle de la feuille de route et de qualité des développements, a entraîné un effet d'éviction sur les développeurs, par ailleurs peu nombreux, issus de la communauté des utilisateurs. Dans le cas de Dokeos, la recherche de rentabilité et d'un time-to-market réduit a également conduit à rompre avec la communauté. Cependant, la communauté apparaît comme une source de résistance au changement parmi d'autres.

Négociation avec la communauté

Pour les trois projets étudiés (Claroline / Claroline Connect, Dokeos et Chamilo), la communauté apparaît comme une force avec laquelle il faut composer (règles) et un ensemble qu'il faut pouvoir canaliser (animation). Il en résulte une lourdeur ainsi qu'un coût pour l'éditeur, en principe compensé par les contributions (promotion du projet, entraide sur les forums, documentation des pratiques, création de modules,...) issues de la communauté. En cas de faibles contributions, le recentrage sur des équipes internes ou apportées par des partenaires, plus facilement contrôlables, peut apparaître comme un choix rationnel. Des efforts en matière de communication et d'animation de la communauté pourraient cependant conduire à une solution plus équilibrée.

Reconfiguration des communautés

Les projets open source suivent généralement à leur création une logique informelle et évoluent ensuite, si nécessaire, vers une autre logique. En pratique, les trois logiques de gouvernance plus matures semblent pouvoir cohabiter sur une même niche fonctionnelle, avec cependant des publics distincts pour communauté. Les entreprises se rassembleraient alors progressivement autour de la logique commerciale (éditeurs et réseau structuré de partenaires) ; les grandes organisations, autour de la logique industrielle (mutualisation dans un cadre coopétitif) et les autres types d'acteurs (p.ex. utilisateurs isolés et très petites entreprises), autour de la logique communautaire (méritocratie et garantie d'équilibre des forces) (Table 3). Si ce mode d'évolution était validé, il annoncerait d'autres mouvements de reconfiguration des communautés au sein de l'écosystème Claroline, autour de 3 projets, incluant Claroline, Chamilo et un troisième projet occupant la place laissée vacante par Dokeos passé en logique propriétaire.

Perspectives

Les codes sources de Claroline, Dokeos, Chamilo et Claroline Connect sont disponibles en ligne. L'activité communautaire peut faire l'objet d'une analyse (métrique) et être comparée projet par projet (p.ex. importance des contributions et ventilation par partenaire). Le site OpenHub fournit des métriques précalculées ainsi que des graphiques. Ces informations n'étant pas disponibles pour Claroline Connect (migration du dépôt de Sourceforge vers Github), un travail supplémentaire de collecte et d'homogénéisation des métriques est donc à prévoir.

Les dimensions de la gouvernance ont été détaillées par Markus (200) et Laffan (2012). Les modalités de gouvernance pourraient ainsi être caractérisées plus précisément, notamment à des fins de comparaison objective.

Le point de vue des utilisateurs a été approché au cours de l'étude, que ce soit par des entretiens plus courts ou l'atelier créatif organisé lors de la conférence annuelle. La réalisation d'interviews d'acteurs ayant migré permettrait d'obtenir un éclairage complémentaire sur les faiblesses du projet Claroline en matière de communication. Le constat d'un manque de communication et d'animation suppose un travail davantage ancré dans la réalité quotidienne des projets pour (1) valider ce constat, (2) proposer des mesures correctives et (3) en tester l'efficacité. Ce travail de recherche-action est une suite possible au traitement complet des entretiens réalisés pour cette étude de cas.

Bibliographie

- Bonaccorsi, A., & Rossi, C. (2006). Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business. *Knowledge, Technology & Policy*, 18(4), pp. 40-64.
- CNLL (2015). Impact du Logiciel Libre / Open Source Software en France 2015-2020 - Quels enjeux de marchés, d'emploi, de formation et d'innovation . Pierre Audoin Conseil, 19 novembre 2015 ; en ligne : <http://cnll.fr/static/pdf/pac-logiciels-libres-2015.pdf>.
- Dahlander, L., & Magnusson, M. G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research policy*, 34(4), pp. 481-493.
- De Laat, P. B. (2007). Governance of open source software: state of the art. *Journal of Management & Governance*, 11(2), pp. 165-177.
- Feller, J. & Fitzgerald, B. (2002). *Understanding open source software development*, Addison-Wesley.
- Fitzgerald, B. (2006). The transformation of open source software. *Mis Quarterly*, pp. 587-598.
- Franke, N., & Von Hippel, E. (2003). Satisfying heterogeneous user needs via innovation toolkits: the case of Apache security software. *Research policy*, 32(7), 1199-1215.
- Giordano, Y. (2003). *Conduire un projet de recherche. Une perspective qualitative*. Editions EMS.

- Laffan, L. (2012). A new way of measuring openness: The open governance index. *Technology Innovation Management Review*, 2(1).
- Lakhani, K., Wolf, B., Bates, J., & DiBona, C. (2002). *The boston consulting group hacker survey*. The Boston Consulting Group.
- MacCormack, A., Rusnak, J., & Baldwin, C. Y. (2006). Exploring the structure of complex software designs: An empirical study of open source and proprietary code. *Management Science*, 52(7), 1015-1030.
- Markus, M. L. (2007). The governance of free/open source software projects: monolithic, multidimensional, or configurational?. *Journal of Management & Governance*, 11(2), pp. 151-163.
- Roberts, J. A., Hann, I. H., & Slaughter, S. A. (2006). Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management science*, 52(7), pp. 984-999.
- Teece, D. J. (2010). Business models, business strategy and innovation. *Long range planning*, 43(2), pp. 172-194.
- Thiéart, R. A. (2007). *Méthodes de recherche en management - 3ème édition*. Dunod.
- Valimaki, M. (2003). Dual Licensing in Open Source Software Industry. *Systèmes d'Information et Management*. Vol. 8 : Iss. 1 , Article 4.
- Vasquez Bronfman, S., Miralles, F. (2007). Business Models in Open Source Software: do they exist?. In *12ème conférence de l'Association Information et Management (AIM)*, Lausanne (Suisse), 18-19 juin 2007.
- Viseur, R. (2007). Gestion de communautés Open Source. In *12ème conférence de l'Association Information et Management (AIM)*, Lausanne (Suisse), 18-19 juin 2007.
- Viseur, R. (2012). Forks impacts and motivations in free and open source projects. *International Journal of Advanced Computer Science and Applications*, 3(2), pp. 117-122.
- Viseur, R. (2013a). Evolution des stratégies et modèles d'affaires des éditeurs Open Source face au Cloud computing. *Terminal. Technologie de l'information, culture & société*, (113-114), pp. 173-193.
- Viseur R., Pinchart L. (2013b). Developing Free Software within a Major ICT Company, *CommEx*, Capodistria (Slovenia).
- Viseur, R., & Robles, G. (2015). First Results About Motivation and Impact of License Changes in Open Source Projects. In *IFIP International Conference on Open Source Systems*, pp. 137-145, Springer.
- Viseur, R. (2016). Gouvernance des projets open source. In *INFORSID*, Grenoble (France), pp. 181-198.
- West, J., & O'mahony, S. (2008). The role of participation architecture in growing sponsored open source communities. *Industry and innovation*, 15(2), pp. 145-168.
- Yin, R. K. (2009). *Case study research: Design and methods*. Sage publications.