Stratégies *open-sources* et innovation de modèle d'affaires: le cas de l'entreprise Dokeos

Robert Viseur*
Nicolas Jullien**
Amel Charleux****
Anne Mione****
* FWEG, Université de Mons, Belgique
** IMT Atlantique, Brest, France
*** Université de Montpellier, France

Résumé :

L'*open source* constitue un cas extrême d'*open innovation*. Si les modèles d'affaires des prestataires *open source* sont aujourd'hui bien connus, la compréhension de la rationalité derrière le choix d'un modèle d'affaires plutôt qu'un autre, et surtout l'analyse de la dynamique des modèles au cours de la vie de l'entreprise, et des projets libres, continue à faire défaut. Notre recherche ambitionne de contribuer à combler ce manque répondre à ce manque à partir d'une étude de cas sur le LMS (*Learning Management System*), et le prestataire éponyme, Dokeos (soit un *fork* du logiciel *open source* Claroline), dont est issu le logiciel Chamilo, tous concurrents du logiciel *open source* Moodle. Nos résultats permettent de mieux comprendre le rôle de la communauté dans le modèle d'affaires des éditeurs *open source* ainsi que la rationalité derrière l'évolution des stratégies de ces éditeurs au travers de la construction progressive d'actifs spécifiques.

Mots clés :

open source, modèle d'affaires, LMS, BMI, BMC.

1. Introduction

Le logiciel libre est un système d'innovation qui a été initié par les utilisateurs (Von Hippel, 2001). Il a rapidement été adopté par des entreprises pour proposer des solutions commerciales appuyées sur un ou plusieurs logiciels libres, stratégies qualifiées d'« *open-source*s » (Fitzgerald, 2006)[1]. S'il existe de nombreux travaux qui étudient ces stratégies (cf. Jullien Viseur 2021 pour une synthèse récente), ils s'accordent à dire qu'ils prolongent des modèles existant en accentuant leur composante « service ».

Parmi ces modèles, celui qui a sans doute été le plus discuté est celui d'éditeur « *open-source* » (Shahrivar et al., 2018 pour une revue récente), car c'est celui qui semble le plus paradoxal, par son abandon, au moins en partie, de la captation de valeur issue de la facturation de l'accès au logiciel (couramment appelée « payement de licence d'utilisation »). Mais l'argument est que l'éditeur est souvent le mieux placé pour facturer un certain nombre de services nécessaires pour l'usage du logiciel, et aussi qu'il bénéficie et qu'il bénéficie de coûts de développement moins élevés dus à la mutualisation des efforts de développement, au signalement d'erreurs de programmation, à l'expression de besoin, à l'exploration de nouvelles pistes par les contributeurs et au *feedback* offert en continu par la communauté à propos des orientations prises ou annoncées (Shahrivar et al., 2018). Il peut ainsi, éventuellement, reconsidérer sa feuille de route (Teigland et al., 2014). La communauté reste donc un soutien important pour l'entretien et l'accroissement de la couverture fonctionnelle ainsi que pour la popularisation de la marque via le bouche-à-oreille. Cependant, en plus de l'absence des revenus liés à la vente de licences d'accès du modèle classique, la gestion de cette communauté peut être coûteuse, notamment si les besoins exprimés sont trop variés pour être pris en compte.

Pourtant, ces communautés ne sont pas dociles. En témoignent les exemples de *forks* de projets de logiciels libres connus (Viseur, 2012). Ces derniers peuvent se produire sur des projets développés dans un cadre commercial (p. ex. *fork* MariaDB de MySQL), ou non (p. ex. Claroline ; cf. Viseur et Charleux, 2019). Parmi les motifs expliquant les conflits, et leur forme la plus extrême, à savoir le *fork*, les changements de stratégie et de modèles d'affaires (p. ex. passage à un modèle de double licence ou à un modèle de services de type SaaS), tout autant que le contrôle des ressources (p. ex. marques) nécessaires à la pérennité de ces derniers, semblent occuper une place importante bien que peu étudiée dans le détail dans la littérature scientifique.

Autrement dit, il s'agit autant d'étudier un modèle d'affaires que la dynamique de ces modèles, qui est elle-même liée à la dynamique des projets de logiciel libre (Jullien et Viseur, 2021). Nous proposons donc d'étudier l'innovation de modèles d'affaires des éditeurs *open source* et des effets induits sur la stabilité des projets dans le cas particulier de l'entreprise Dokeos, dont le logiciel est issu d'un *fork* du logiciel Claroline et lui-même forké pour donner la naissance au logiciel Chamilo.

---

1 Suivant cet auteur, nous utiliserons le terme « logiciel libre » pour parler des dimensions technico-organisationelle (développement ouvert et collectif d'un logiciel) et produit (le logiciel), et le terme « open source » pour parler des dimensions et stratégies marchandes du logiciel libre.

Cet article sera organisé en trois parties. Dans une première partie, nous procéderons à un rapide état de l'art en matière d'innovation de modèle d'affaires, avant de particulariser la réflexion au secteur *open source* et de présenter les conflits générés au sein des communautés *open source* suite à des changements de stratégie ou de modèle d'affaires. Dans une seconde partie, et après avoir présenté notre méthodologie à base d'études de cas, nous présenterons nos résultats. Dans une troisième partie, et avant de conclure, nous discuterons nos résultats, en particulier les questions du rôle de la communauté dans les projets de logiciels libres, du cycle de vie des modèles d'affaires *open source* et de la construction des actifs spécifiques.

2. État de l'art

2.1. Modèles économiques du libre, et modèles d'affaires open-source

D'après Massa et al. (2017), qui proposent une revue de littérature sur le cadre analytique du modèle économique, celui-ci améliore les cadres précédents (p. ex. Ressource Based View), car il prend en compte trois aspects pas ou mal pris en compte par ceux-ci : le fait que la valeur est souvent créée par un réseau d'acteurs (Lepak et al., 2007, p. 513), ou un « écosystème » (Adner et Kappor, 2010), et non pas le résultat de l'action d'un seul acteur économique (notamment dans le numérique, cf. Amit et Zott, 2001), même si un acteur est souvent au centre de ce réseau (cf. la recherche sur les plateformes numériques, résumée par Reuver et al,, 2018, qui est un cas particulier de plateforme technologique, telle qu'étudiée par Wareham et al., 2015) ; que ces acteurs construisent des relations en situation d'asymétrie d'information sur leurs besoins et leur contrôle de la technologie ; et que par conséquent création et captation de la valeur sont deux choses différentes, pas forcément réalisées de façon liée par les acteurs. Ces trois éléments apparaissent au cœur du fonctionnement de l'industrie du logiciel (et partant, de l'*open source*).

Ainsi, la difficulté d'un éditeur *open source* réside souvent dans sa difficulté à capter suffisamment de valeur du projet qu'il anime (Massa et al., 2017, p. 89 et suivantes discutent les liens entre modèle économique et stratégie). Il faut trouver le bon équilibre entre une organisation hiérarchique, source d'efficacité productive (Lee et al., 2017) et de plus grandes possibilités de captation de la valeur, et l'ouverture, qui facilite l'intégration de nouveaux membres et donc l'innovation, la création de valeur. Un dilemme finalement assez classique entre exploration et exploitation d'une solution technologique (March, 1991), notamment dans l'économie des plateformes (en plus des références déjà mentionnées, on pourra consulter Wareham et al., 2014 et Gawer, 2014).

Selon Teece (2018), cela veut aussi dire qu'il faut assurer l'alignement interne et la cohérence des éléments de captation de valeur, ce qu'il appelle les modèles de revenus (incluant la tarification, les canaux et le type d'interaction avec la clientèle ciblée) et les coûts de production des activités, des ressources clefs, et de contrôle des partenaires clefs c'est-à-dire, pour reprendre Dierickx et Cool (1989), des capacités difficilement échangeables ou imitables. Dans notre cas, ces coûts inclus la gestion du projet libre, qui est la base de la création de valeur, et les autres actifs qui permettent de créer des valeurs complémentaires et surtout d'en capter une partie. Six types d'actifs spécifiques ont ainsi été identifiés : des actifs humains, des actifs technologiques, des actifs organisationnels, internes et externes, des actifs réputationnels et des actifs commerciaux.

L'importance des collaborations dans les modèles d'affaires des entreprises n'est pas propre au logiciel.. Laudien et al. (2017) soulignent par exemple la transformation du consommateur

passif en co-créateur de valeur. Plus largement, les pratiquées liées à l'*open innovation* engagent les entreprises dans des processus de création et de capture de valeur impliquant différentes parties prenantes. À la conception classique *inside in* / *inside out* de Chesbrough (2003), a succédé une vision plus complexe des activités de création et de captation de valeur (Chesbrough et al., 2018) vues non seulement comme un stock (concept de « *value-in-use* ») mais aussi comme un flux (concept de « *value-in-exchange* ») que l'entreprise peut alimenter et/ou capter dans un environnement fortement coopétitif. Il nous semble que le fonctionnement des écosystèmes *open source* permet d'explorer ce cadre. En croisant ces deux types de valeur avec la création et la captation de valeur, ils distinguent quatre quadrants (cf. Tableau 1) que nous allons maintenant relier au contexte des stratégies *open source*.

Tableau 1 : valeur en *open innovation* chez Chesbrough et al. (2018)

	Valeur dans l'usage	**Valeur dans l'échange**
Création de valeur	Réalisation de valeur	Provision de valeur
Captation de valeur	Exploitation de la valeur	Négociation de valeur (appropriabilité)

La création de valeur se produit au sein du projet libre et résulte de l'accès ouvert à un stock (un logiciel), mais surtout à l'entretien constant de ce logiciel, via un flux entretenu de contributions des membres, utilisateurs, individus ou entreprises, ou, ce qui nous intéresse plus ici, entreprises *opensources,* c'est-à-dire des entreprises qui cherchent à capter une partie de la valeur créée par le projet. Le processus de création de valeur dominant est donc la provision de valeur (c'est-à-dire la mise à disposition de ressources à des partenaires qui vont évaluer ces ressources en vue d'un usage futur). Une fois de plus, cette provision de valeur n'est pas simplement altruiste, c'est une forme de contrôle, par le déploiement de ressources (Schaarschmidt et al., 2015) d'un projet car, toujours en s'appuyant sur le cadre plus général de l'économie des plateformes, « *the technology is at the core of the monitoring of the tasks, it is central in the governance [of a digital platform]* » (Mini et al. 2021).

Dans ce contexte, la captation de valeur du projet peut bien sûr être réalisée par le ou les créateurs de valeur. La maîtrise de la technologie consécutive à leur engagement dans le projet les mettra d'ailleurs dans une position avantageuse pour monétiser l'accès au logiciel (logiciel stock, position de l'éditeur), mais aussi, surtout, l'accès à des services complémentaires, permettant de gérer le logiciel flux pour des clients, qui n'en n'ont pas les compétences ou l'intérêt économique (assurance, adaptation ; Jullien et Zimmermann, 2011). Ces stratégies open-sources nécessitent souvent la création d'actifs stratégiques propres, sous forme de propriété intellectuelle logicielle (module spécifique) ou d'infrastructure matérielle (SaaS) ou d'outils spécifiques rendant l'exploitation du logiciel plus efficace (p. ex. scripts d'intégration et de migration). Autrement dit, un projet logiciel libre fonctionne s'il arrive à laisser les utilisateurs exploiter une partie de la valeur créée par l'écosystème, tant qu'ils participent à son attractivité (diffusion de la technologie.) Cette provision de valeur pouvant être partagée (contribution au projet) ou réalisée en interne. Pour qu'on projet fonctionne, il faut qu'il soit suffisamment dynamique pour obliger certains acteurs à participer à la provision de valeur partagée (contribution au logiciel-flux).

Pour les participants, et c'est le deuxième point du modèle d'innovation ouverte de Chesbrough et al. (2018) , il s'agit donc d'un modèle qui repose sur l'optimisation, plutôt que de maximisation de la captation de valeur immédiate. L'entreprise qui anime la plateforme technologique, dans notre cas l'éditeur open-source, ou entreprise « focale » dans le modèle de Chesbrough, doit tendre vers un équilibre entre ses propres intérêts et ceux de ses « partenaires » dans un contexte d'*open innovation*, le contexte libre ayant cette spécificité supplémentaire qu'il n'y a pas de sélection a priori des-dits partenaires du fait de la nature des licences (Demil et Lecocq, 2003). Sinon le logiciel-flux meurt progressivement et (re)devient alors logiciel-stock car il n'y a plus d'innovation ouverte, voire plus d'innovation. C'est ce qui est arrivé au projet autour du système d'exploitation Symbian, par exemple (Fautrero et Gueguen, 2012).

Cependant, cela peut être un choix délibéré. Osterloh et Rota (2007) rappellent que l'ouverture rencontrée dans le logiciel libre, comme dans l'innovation en général, tend à n'être qu'une situation transitoire permettant de faire évoluer plus rapidement la technologie dans un contexte d'incertitude. L'innovateur se réserve généralement dans un second temps les fruits de ses contributions. Nous traduisons cela dans le logiciel libre, et suivant Jullien et Viseur (2021), par le fait que lorsque la solution logicielle devient mature, c'est-à-dire que le processus de développement ralentit, se concentre sur la maintenance de l'existant (du logiciel-stock), l'éditeur peut être tenté de recentrer son modèle d'affaires, qui était basé sur la gestion du logiciel flux, sur la gestion d'un logiciel stock et à se rapprocher des modèles classiques. Cela peut aussi être compris dans l'autre sens : parce que l'éditeur souhaite se concentrer sur des modèles de captation basés sur le logiciel stock, il va diminuer ses investissements dans le projet libre, dans la dynamique d'innovation, dans l'animation de l'écosystème communautaire.

Autrement dit, l'objectif de cet article est d'étudier comment les éditeurs open source arbitrent entre le modèle économique du projet, et leur modèle économique, que nous nommerons par la suite « modèle d'affaires » (pour bien distinguer le modèle économique de l'ensemble de l'écosystème du modèle de l'éditeur qui le porte). Ces modèles d'affaires, par leurs caractéristiques, peuvent être plus ou moins favorables à la coopération de l'éditeur open-source avec son écosystème, parce que plus ou moins dépendants des contributions en termes de captation d'une création de valeur (gestion du logiciel flux) ou de la gestion de l'accès au logiciel stock, générant plus ou moins de tensions, et des réactions différentes des contributeurs.

Tableau 2: modèles d'affaires et logiques de gouvernance (adapté de Jullien & Viseur, 2021; de Viseur & Charleux, 2019; et de Charleux & Mione, 2018)

Prestataire *open source*				
Types de modèles d'affaires	engagement	expertise	exploration	exploitation
Phases	1	1, 2	2, 3	2, 3
Modèles	personnalisation	personnalisation, intégration...	mutualisation (bien commun industriel et bien communs	édition (p.ex. double licence et

d'affaires			numériques)	*open core*)
Exemples	*freelances* spécialisés *open source*	sociétés de service en logiciel libre	grandes entreprises (p.ex. IBM: Eclipse) et collaboration/coopétition d'entreprises (p.ex. Apache)	MySQL, Odoo
Projet libre				
Niveaux de contrôle	variable	modéré à fort	modéré à fort	fort
Logiques de gouvernance	informelle	commerciale ou industrielle	communautaire ou industrielle	commerciale
Exemples	Gimp	Tryton (B2CK), Linshare (Linagora)	Apache, Mozilla (Firefox), PostgreSQL, OpenOffice.org, LibreOffice.org	Alfresco, Elasticsearch, MySQL, Odoo

Viseur et Charleux (2019) ont montré le nécessaire alignement entre la stratégie de l'éditeur, son modèle d'affaires et la gouvernance du projet libre (cf. Tableau 2). Les modèles d'affaires *open source* peuvent se classer entre modèle d'engagement (basé sur une collaboration informelle et adapté au modèle d'affaires de personnalisation), d'exploration (basé sur la mise en commun des ressources sous l'égide d'un tiers de confiance tel qu'une fondation), d'expertise (basé sur la prestation de services et adapté aux modèles d'affaires de personnalisation ou d'intégration) et d'exploitation (davantage basé sur l'édition logicielle) (Charleux et Mione, 2018 ; Jullien et Viseur, 2021). Suivant le modèle d'affaires pratiqué, un contrôle plus ou moins fort du projet libre par le prestataire sera nécessaire. Ainsi l'éditeur open-source cherchera souvent à étroitement contrôler le projet. Il adoptera donc plutôt une logique commerciale, là où l'intégrateur préférera sans doute une logique communautaire gage de l'indépendance du projet libre. Le consortium industriel cherchant à partager les coûts mais souhaitant garder un contrôle de la feuille de route choira par contre une logique industrielle permettant d'affaiblir les contributeurs agissant hors consortium.

Ici, nous voulons étudier pourquoi ils font parfois évoluer leur modèle et comment cette évolution, que nous appellerons « innovation du modèle d'affaires » est négociée avec la communauté des utilisateurs-développeurs, acceptée ou non par celle-ci, et quelles en sont les conséquences pour l'éditeur. Cela passe d'abord par circonscrire ce qu'est une innovation de modèle d'affaires, avant de discuter ce que nous entendons par négociation.

2.2. L'innovation de modèle d'affaires

Laudien et al. (2017) différencient les innovations de modèle d'affaires des ajustements de modèles d'affaires. La première apporte des changements radicaux dans le modèle d'affaires de la firme ou une nouveauté à l'échelle d'un secteur industriel tandis que la seconde amène un changement incrémental. Cette distinction se retrouve chez Foss et Saebi (2017), l'ajustement de Laudien et al. (2017) s'assimilant au mode évolutionnaire de Foss et Saebi (2017). Ces derniers proposent une typologie plus fine des innovations de modèles d'affaires en fonction, d'une part, de leur nouveauté (nouveauté pour la firme ou pour le secteur) et de la portée (impact sur un composant du modèle d'affaires ou sur plusieurs composants). Laudien et al. (2017) constatent également qu'un modèle d'affaires traverse, comme un produit ou une marque, un cycle de vie. Ils identifient quatre phases : la phase d'invention, la phase d'innovation, la phase d'ajustement et la phase de retrait. Les deux premières sont davantage marquées par des logiques d'exploration tandis que les deux autres le sont surtout par celles d'exploitation.

Cela met en exergue différents facteurs d'innovation de modèle d'affaires dans le secteur open source. Ainsi des innovations technologiques (nouveauté pour le secteur informatique), la dernière étant sans doute le cloud computing (entraînant le développement des offres SaaS), ont entraîné des évolutions fortes du modèle d'éditeur open-source. Dans ce cas, le projet et l'innovation qui étaient au cœur du modèle d'affaires (captation de valeur basée sur le suivi de l'innovation) deviennent secondaires, l'entreprise cherchant d'abord à disposer d'une solution stable (fiable et sans trop de changement de fonctionnalités) pour proposer l'accès à une « capacités techniques entretenues » (Viseur, 2013a). Cependant, les modèles d'affaires sont surtout liés à la phase du projet logiciel libre (initiation d'un projet, phase de développement et structuration, phase de maturité). Il s'agit donc d'ajuster son offre et ses capacités à la dynamique du projet et aux opportunités commerciales, de captation de valeur, générées. La capacité d'un acteur open-source, et notamment de l'éditeur, à faire évoluer son modèle avec le temps et les phrases suppose alors la pré-existence de certains actifs spécifiques et l'allocation de ressources pour la constructions de nouveaux (Laudien et al. 2017).

2.3. Éléments de tensions, éléments de réaction

Comme dans tout groupe, des conflits existent, qui peuvent être de nature intra-organisationnelle (désaccord entre des personnes, par exemple) et le projet, pour réussir, doit être capable de mettre en place des structures de gouvernance capables de les gérer (van Wendel de Joode, 2004 ; Jensen et Scacchi, 2005 ; O'mahony et Ferraro, 2007). Ces conflits peuvent être aussi l'indice de divergence stratégiques entre les participants (ibid). C'est ce que Filippova et Cho (2016) ont appelé un conflit de norme. Ils soulignent que c'est le seul élément qui semble corrélé avec un désengagement de contributeurs. C'est ce qui va nous intéresser particulièrement ici : quand une décision de l'éditeur, qui est aussi celui qui gère le projet, engendre des réactions d'un ou plusieurs contributeurs, au moins des désaccords exprimés, au plus des départs de certains.

Parmi les perturbations induites, citons les changements de licences, car ces dernières « *fournissent une sorte de "constitution" ou d'accord juridique sur la manière dont le projet est développé et distribué* » (Viseur & Robles, 2015). Si ces changements sont parfois dus à des contraintes techniques (Di Penta et al. 2010), de nombreux auteurs on soulignés le lien

entre changement de licence et de stratégies ou de modèles d'affaires (Onetti et Verma, 2009 ; Viseur, 2013a). Cependant, d'autres éléments peuvent engendrer des conflits, comme la façon dont les contributions sont acceptées (qui a le droit de contribuer à la version « officielle »), des décisions sur l'architecture du logiciel, qui facile, plus ou moins la contribution selon son niveau de modularité et de clarté, sur la feuille de route du projet qui ne prennent pas en compte des demandes, etc.

Ces conflits se traduisent par différentes manifestations analysées par Viseur et Charleux (2021), à la suite d'autres auteurs comme Wendel de Joode (2004), à travers le modèle d'Hirschman (2017) « *exit, voice, loyalty* » et son extension par Rusbult et al, (1982), qui rajoutent la négligence, ou l'apathie. Ce modèle s'intéresse d'abord au comportement individuel face à un pouvoir, surtout organisationnel, et a été beaucoup utilisé en théories de l'organisation et notamment dans l'étude des relations entre employeur et employés, depuis Farrel (1983). Notre approche est un peu différente, puisque s'il y a bien un pouvoir organisationnel, l'éditeur open-source, c'est plutôt la capacité des contributeurs à réagir collectivement à ces décisions. Cependant, il reste de ces travaux, et notamment des analyses de Viseur et Charleux (2019, 2021) que les réactions sont médiatés par la plate-forme, qui permet aussi aux contributeurs de se coordonner et de peser face à l'éditeur. Autrement, dit, il y a une gradation des réactions face à une situation insatisfaisante, qu'elle ait été provoquée par une décision spécifique de l'éditeur, à un moment précis, comme un changement de licence, ou par un comportement, une attitude qui engendre un mécontentement plus diffus, qui met plus de temps à se cristalliser, mais qui peut être aussi impactant pour le projet s'il aboutit à un départ massif et coordonné des contributeurs, vers un nouveau projet (ce qu'on appelle un « *fork* »).

Bref, il s'agit d'étudier les épreuves que traverse un projet logiciel libre géré par un éditeur open-source, c'est-à-dire de détecter les éléments de mécontentement exprimés et les actions entreprises dans l'histoire du projet. En menant une étude de cas sur l'écosystème Dokeos, nous cherchons à construire des éléments de généralisation analytique (Gerring 2007 ; Lee et Baskerville 2003) des éléments déclencheurs et des réponses collectives des conflits entre un éditeur *open-source* et sa communauté de développement.

3. Données et méthodologie

La recherche s'appuie sur l'étude de l'écosystème Claroline au sens large, à savoir les logiciels Claroline (version 1 et version 2), le logiciel Dokeos et le logiciel Chamilo. Dans cette étude de cas, nous avons choisi une approche par entretien essentiellement, pour aller au-delà des éléments exprimés dans les forums, ou liste de diffusion des projets et disposer de la réflexion des acteurs, des éléments de justification de leur décision. Cette posture ne nous met pas à l'abri d'une ré-écriture de l'histoire par ces acteurs au prisme de leur situation actuelle. C'est pourquoi les informations collectées l'ont été à différents stades de l'histoire des projets, avec notamment la ré-utilisation d'entretiens réalisés. La cohérence interne des explications et analyses que les mêmes acteurs donnent dans les entretiens plus récents a ainsi été vérifiée.

Précisément, nous nous sommes appuyés sur le matériau collecté dans une première étude de cas réalisée en 2006 spécifiquement autour de Claroline, sur base d'un questionnaire rempli par le principal développeur du projet (quatorze pages de réponses) (Viseur, 2007). À la suite de cela, huit interviews, sur base d'un guide d'entretien semi-directif, d'une durée comprise

entre 1:39:08 et 2:35:19, ont été conduites auprès de responsables et de développeurs de ces trois projets. Ces interviews ont été enregistrées, ont fait l'objet d'une importante prise de note et d'une retranscription suivie d'un codage. Une présentation de résultats préliminaires a été réalisée lors de la conférence annuelle en 2017 tandis qu'un atelier pour redynamiser la communauté Claroline a été réalisé dans un second temps histoire de disposer d'un point de vue plus proche de celui des utilisateurs. Deux interviews ont été refaites fin 2021 auprès des fondateurs de Dokeos (2:04:47) et de Chamilo (2:51:23) pour approfondir les évolutions récentes du modèle d'affaires de Dokeos et les conditions de survenance du fork.

4. Résultats

Le logiciel Dokeos, et la société éponyme, a été créé par Thomas de Praetere en 2003, en tant que *fork* du logiciel *open source* Claroline, dont il est également co-fondateur. Nous présentons dans cette section l'historique du prestataire Dokeos, ses évolutions stratégiques et la place de la communauté dans le projet éponyme.

4.1. Les différentes épreuves et leurs conséquences

L'innovation apportée par le logiciel Claroline démarre en 2001 à l'initiative d'utilisateurs de pointe, employés par l'Université Catholique de Louvain (UCL), confrontés à un besoin en avance sur ceux de leurs pairs, dès lors poussé à innover par eux-mêmes. Le projet Dokeos est créé suite à un désaccord entre le *leader* du projet libre (Claroline) et le sponsor (UCL). Cet utilisateur de pointe souhaite devenir utilisateur-entrepreneur (Baldwin, Hienerth & Von Hippel, 2006). Le conflit se cristallise autour du contrôle de la marque, propriété de l'UCL, un motif fréquent de *fork* (Viseur, 2012). Il est bref et violent, ne permet donc pas l'émergence d'un compromis, se solde en 2003 par une rupture (*fork*) et la création du prestataire Dokeos ainsi que du logiciel éponyme. Dokeos est donc créé pour pouvoir offrir des services sur le LMS. Ce modèle d'affaires évolue rapidement d'un modèle de personnalisation vers un modèle d'édition *open source* (Jullien et Viseur, 2021).

La seconde moitié de l'année 2009 est marquée par deux évènements. D'une part, la relation avec un partenaire commercial en charge de développements (sous-traitance) se dégrade suite à l'impossibilité de tenir les délais vis-à-vis d'un client. D'autre part, la volonté de privatiser certains modules, soit un modèle de dual licensing, entraîne de nouvelles tensions avec ce même partenaire. Ces tensions conduisent au fork Chamilo, lequel est suivi par la communauté. La sortie du modèle open source s'accompagne en 2020 de la réécriture du logiciel, d'une part, en vue de faire du SaaS (Software as a Service), d'autre part, de faire évoluer les bases technologiques (Ruby on Rails plutôt que PHP).

Après 2020, l'entreprise Dokeos commercialise donc une solution hébergée sous la forme d'un abonnement. Le logiciel n'est plus téléchargeable. Une activité d'accompagnement en projet d'elearning est proposée dans un second temps. L'entreprise se trouve alors confrontée à une nouvelle concurrence, celle des startups très bien capitalisées capables de croître sans nécessité de rentabilité à court terme.

4.2. Les raisons stratégiques de l'évolution du modèle d'affaires

La décision de créer le projet Dokeos découle de l'opportunité de développer une activité de services autour du logiciel Claroline et de l'incapacité à dégager un accord entre le sponsor et

l'entrepreneur, alors employé du sponsor, autour du contrôle de la marque. Le projet est donc forké, et son nom, modifié.

> «*On est pas du tout sur le marché de l'enseignement logiciel et, nos conclusions, c'est qu'il n'y avait pas de marché en fait. C'est-à-dire que les écoles disposent de main d'œuvre gratuite on va dire... en tout cas financée par le denier public. Et, du coup, c'est toujours moins cher pour elles de le faire elle-même que de l'acheter.*»

Rapidement, la clientèle de Dokeos évolue par rapport à la cible initiale. Le prestataire se détourne du marché de l'enseignement suite à l'échec de la vente de services aux universités. Il s'oriente vers une niche rentable constituée par le marché du médical en général et le marché pharmaceutique en particulier. Les besoins de ces nouveaux clients sont spécifiques et entraînent de nouveaux développements en matière de *reporting*, de *blended learning* et d'*unboarding*. Le projet libre, majoritairement porté par des développeurs issus d'universités et de hautes écoles, devient progressivement moins important compte tenu des nouvelles fonctionnalités demandées par les clients, c'est-à-dire les utilisateurs ayant des besoins métiers spécifiques et prêts à payer.

En contrepartie de cette divergence avec la communauté, l'entreprise tente de créer de la propriété intellectuelle en privatisant certains modules, ce qui ouvre un conflit avec un partenaire proche de la communauté. En résulte le fork Chamilo, suivi par la communauté, permettant au prestataire focal, Beeznest, de préserver la neutralité du projet libre, protégé par la mise en association. Ce conflit ouvert s'accompagne d'un problème de réputation suite à la rédaction d'un article[2] dans la presse française (« Quand un conflit se sait, tu perds en crédibilité en fait »). Soucieux de contrôler le développement de sa marque ainsi que sa chaîne de valeur, le prestataire Dokeos décide de sortir du modèle open source.

La décision de re-développement s'explique par l'âge du développement logiciel et la nécessite de repartir sur des bases saines pour des raisons de complexité, de sécurité et d'interopérabilité du logiciel. L'ordre de grandeur de l'investissement est de un million d'euros auquel s'ajoute deux à trois cent mille euros annuels pour l'adaptation du logiciel aux demandes des clients. Autant que le fork et les opportunités liées au SaaS, l'obsolescence technologique conduit donc au re-développement du logiciel (« à un moment, le logiciel est vieux »). La société Belighted soutient cette transformation, laquelle impacte également les méthodes de développement au sein de l'entreprise, modernisées. De manière à maintenir un socle fonctionnel étroit, et ainsi maîtriser les coûts de développement, l'entreprise choisit de développer l'interopérabilité, via des APIs ouvertes, avec des solutions techniques complémentaires : Enveo (gestion administrative), HP5 (outils d'auteurs), 8x8 (visioconférence), Wordpress (e-commerce)… (soit environ 30 autres logiciels compatibles au final).

Le modèle SaaS s'accompagne cependant d'une attrition non négligeable liée au manque d'accompagnement des clients. L'entreprise développe donc une activité de conseil, elle-aussi facturée, permettant donc d'accroître les revenus mais aussi de fidéliser la clientèle du SaaS. L'entreprise, toujours pilotée depuis la Belgique, dispose alors d'une première filiale en Inde pour l'activité de développement et d'une seconde au Maroc pour l'activité de conseil. En valeur relative, les investissements dans le logiciel diminuent, ce qui est justifié

2 Cf. http://www.e-learning-infos.com/info_article/m/931/mesaventure-dun-lms-open-source%E2%80%A6.html.

par la « commoditisation du logiciel ». Si la technologie n'est pas publiée à nouveau en open source, c'est essentiellement pour ne pas armer la concurrence, à même dès lors de proposer des hébergements à prix cassé.

L'émergence d'une nouvelle concurrence (startups en elearning) et le succès du projet libre Moodle suscitent des réflexions quant à l'intérêt d'abandonner le développement d'une solution maison coûteuse sur le plan de la maintenance. La politique actuelle d'un développement interne est justifiée, d'une part, par la fragilité des modèles d'affaires basés uniquement sur des capacités humaines (Gadray, 2003), disposant de peu de revenus récurrents pour amortir d'éventuelles variations conjoncturelles de chiffre d'affaires, d'autre part, par la vétusté de Moodle sur le plan du design des interfaces graphiques, un critère de choix important pour le segment de clientèle visé.

4.3. Place et réaction de la communauté

Par rapport à un développement interne classique (traditionnellement, l'entreprise peut acheter[3], faire ou faire faire), le modèle d'affaires de l'UCL concernant Claroline s'appuie sur la communauté open source en tant que partenaire clef (PC) de manière à mutualiser les efforts de développement (exploration et production de code source fonctionnel). L'open source ajoute donc le « faire avec » compte tenu du support de la communauté. Une grande partie de la communauté suit le co-fondateur du projet libre Claroline lorsqu'il fonde le projet Dokeos et l'entreprise éponyme. Au début de la vie du projet Dokeos, la communauté fournit un complément de ressources à l'entreprise pour stimuler le développement du logiciel. Elle apporte en outre un buzz positif à l'entreprise éponyme. L'utilité de la communauté se réduit par contre au fil du temps. En effet, elle se compose majoritairement de développeurs issus d'institutions d'enseignement public supérieur ou universitaire. Ces développeurs promeuvent une logique communautaire pour la gouvernance de cette communauté : la feuille de route du logiciel devrait être décidée par les développeurs sur une base démocratique. Cependant, dans l'intervalle, la clientèle de Dokeos a évolué. Les besoins exprimés par les contributeurs et ceux des clients se mettent à diverger (cf. verbatim), il y a une « opposition de modèle », soit un désalignement progressif entre le modèle d'affaires du prestataire (segment de clientèle) et la gouvernance du projet libre (contenu de la feuille de route souhaité par la communauté).

> «*Et moi, à ce moment-là, je commençais à avoir mes 3, 4, 5 premiers clients qui avaient aussi leurs desiderata et qui disaient, moi je voudrais des fonctionnalités A, B, C. Et puis le jeudi après-midi, j'allais à ma réunion [avec les contributeurs] où ils disaient: moi, je veux des fonctionnalités D, E, F et c'était pas les mêmes. Je proposais celles de mes clients et ils disaient non, la démocratie a voté que non. Oui, mais moi, je sais pas faire autrement. Je suis dans une logique commerciale, je dois faire ce que les gens me disent quoi en fait. Mes patrons, c'est mes clients quoi. J'en ai pas d'autres en fait.*»

En pratique, la communauté n'est dès lors plus gérée par l'éditeur du logiciel Dokeos. Ce rôle se retrouve de facto endossée par un partenaire (sous-traitant). Dès lors que le logiciel est forké, la communauté suit naturellement ce sous-traitant. Pour Dokeos, le partenaire clef

3 À cette époque, WebCT dominait le marché des LMS. La société fut fondée en 1997 et rachetée par Blackboard en 2006, qui vend une solution basée sur Moodle « *Moodlerooms* » selon une logique fortement propriétaire. Blackboard n'est d'ailleurs plus un partenaire certifié Moodle.

n'est plus la communauté mais devient un partenaire technique capable de faire évoluer leur base technologique et leurs méthodes de développement. Par la suite, les partenariats avec d'autres prestataires sont multipliés dans le cadre de ce que l'entrepreneur appelle le « *modèle bavarois* » (Dokeos fonctionne comme « *membre d'un cluster d'innovation* »).

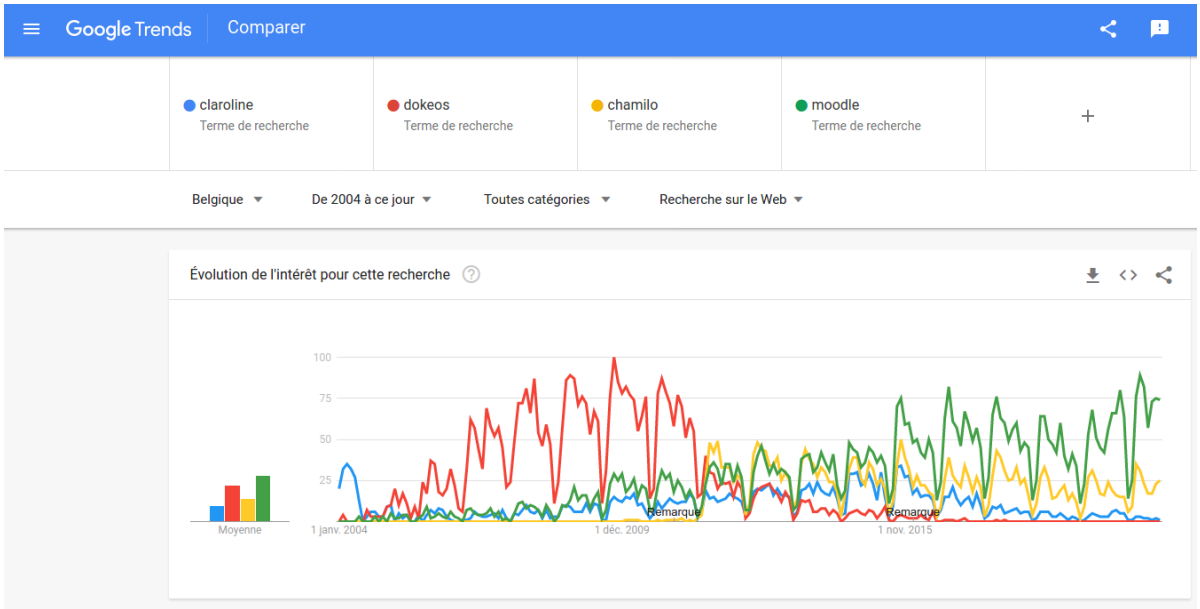
5. Analyse des résultats

Nous analyserons ici l'évolution de la concurrence puis celle du modèle d'affaires de l'entreprise Dokeos.

5.1. Évolution de la concurrence

Ce que nous appellerons écosystème Claroline au sens large (Claroline, Dokeos et Chamilo) a été confronté à la montée de la concurrence d'un autre LMS *open source* : Moodle. Cette nouvelle concurrence offre une alternative supplémentaire lors d'une décision d'adoption d'un LMS ainsi qu'une porte de sortie pour les utilisateurs insatisfaits à la recherche d'une solution alterantive vers laquelle migrer. Moodle est un logiciel de LMS *open source* (sous licence GPL) démarré en tant que projet de recherche en 2001 (Alier et al., 2010). Le travail de thèse associé ne fut jamais achevé car le créateur fut dépassé par le succès du projet *open source*. Moodle connut une forte croissance à partir de mi-2003, encore accélérée à partir de mai 2005. Il est considéré comme mature dès la version 1.9 (2008). Cela se traduit notamment par un *refactoring* du code avec la 2.0 (2010), et la mise à disposition d'une API, après 16 mois de développement.

Figure 1 : développement de la concurrence



Dokeos, apparu en 2003, est confronté à plusieurs concurrences, d'une part celle de Moodle, d'autre part celle de Claroline, dont il est issu, puis celle de son *fork* Chamilo, survenu en 2010. L'écosystème Claroline au sens large est donc affaibli par les *forks* successifs, conduisant à une fragmentation de la communauté et des capacités de développement, alors que son principal concurrent *open source* (Moodle) monte en puissance. Ce que montre l'outil Google Trends (Belgique), c'est d'ailleurs un effondrement de la notoriété de

Claroline au profit de Dokeos puis une montée en puissance de Chamilo dès la survenance du *fork* tandis que Moodle connaît une croissance régulière (cf. Figure 1). Au niveau mondial, la domination de Moodle est encore plus franche.

Reste que la concurrence entre Dokeos et les autres projets liés se révèle moindre qu'on pourrait le croire. En effet, Dokeos se détourne rapidement du marché de l'enseignement, jugé insuffisamment solvable, et adopte rapidement une stratégie de spécialisation vers les marchés pharmaceutique et médical. Une fois passé au modèle SaaS, la concurrence évolue et prend la forme de startups en elearning, massivement capitalisées, face auxquelles Dokeos résiste par l'adoption d'un modèle en réseau, lui permettant de monter la qualité de services et de se centrer sur son cœur de fonctionnalités (cf. Tableau 4). L'impact des forks porte donc moins sur un émiettement du marché que sur la dégradation de la réputation des entreprises touchées (marques) et la fragmentation de la communauté (ressources).

5.2. Modèle d'affaires de Dokeos

Dokeos est donc créé pour pouvoir offrir des services sur le LMS. En 2009, peu avant le fork, l'entreprise vit d'ailleurs toujours d'un modèle d'affaires de personnalisation : un contrat de plusieurs centaines de milliers d'euros concerne en effet cette année-là des développements sur mesure. Ce modèle d'affaires évolue progressivement vers un modèle d'édition open-source. Ce dernier se renforce en 2010 avec la privatisation (dual licensing) de certains modules. Le rôle de la communauté s'amenuise au fil de temps, du fait de la divergence d'objectifs entre les membres du projet libre et le prestataire, au profit d'un partenaire en charge de certains développements. Le Tableau 3 représente le premier modèle d'affaires de Dokeos à travers le canevas proposé par Osterwalder et Pigneur (2011).

Tableau 3 : premier modèle d'affaires (BMC) de Dokeos

<table>
<tr><td rowspan="2"><u>PC</u>
Communauté open source
Entreprise du numérique libre partenaires</td><td><u>AC</u>
Développement</td><td rowspan="2"><u>PV</u>
LMS open source fortement personnalisable
Prestation (installation sur site, intégration, personnalisation...)
Modules propriétaires</td><td><u>CR</u>
Self service
Gestionnaire de compte</td><td rowspan="2"><u>CS</u>
Entreprises</td></tr>
<tr><td><u>RC</u>
Marque</td><td><u>CX</u>
Site Dokeos</td></tr>
<tr><td colspan="2"><u>C€</u>
Développement
Marketing</td><td colspan="3"><u>R€</u>
Gratuit
Prestations de services
Licences (double licence, modules propriétaires)</td></tr>
</table>

Le passage à la double licence, perçue comme une forme de réversion (Chesbrough et al., 2018), suscite la colère de la communauté, et d'un partenaire (ESL) de l'éditeur, laquelle débouche sur le fork Chamilo, structuré par le biais d'une association chargée d'en assurer l'indépendance, soutenu par un prestataire (Beeznest) détenu par le développeur à l'origine du fork. Le modèle d'affaires de Beeznest reprend le modèle d'affaires original de Doekos.

Le modèle d'affaires de Dokeos évolue pour sa part vers un modèle d'affaires de servitisation (offre SaaS).

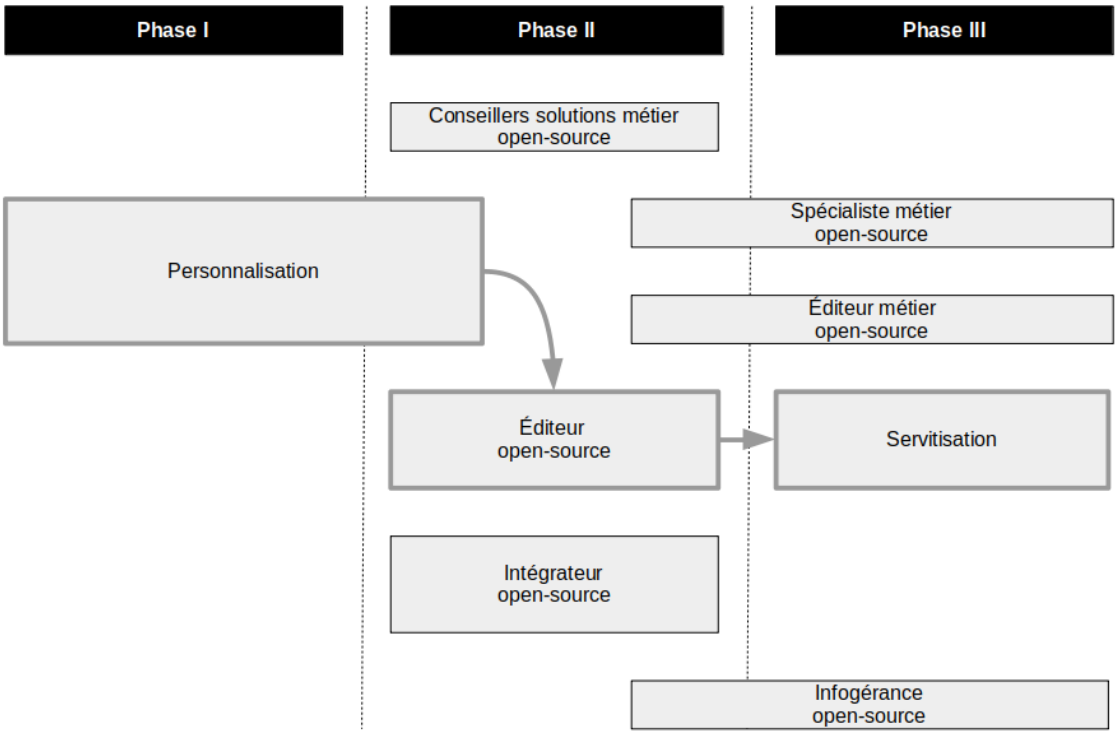
Tableau 4 : deuxième modèle d'affaires de Dokeos après le fork Chamilo

<table>
<tr>
<td rowspan="2"><u>PC</u>
Partenaire technologique (SaaS)
Fournisseurs de logiciels complémentaires (API)</td>
<td><u>AC</u>
Développement
Gestion d'infrastructure</td>
<td rowspan="2"><u>PV</u>
LMS propriétaire en mode SaaS
Support aux utilisateurs (conseil en elearning)</td>
<td><u>CR</u>
Self service
Gestionnaire de compte</td>
<td rowspan="2"><u>CS</u>
Entreprises</td>
</tr>
<tr>
<td><u>RC</u>
Marque
Logiciel</td>
<td><u>CX</u>
Site Dokeos</td>
</tr>
<tr>
<td colspan="3"><u>C€</u>
Développement (Belgique et Inde)
Support (Maroc)
Marketing</td>
<td colspan="2"><u>R€</u>
Abonnements</td>
</tr>
</table>

Dans ce modèle de servitisation, la communauté open source n'est plus un partenaire clef au contraire, d'une part, d'un partenaire technique permettant d'augmenter la maturité des pratiques de développement informatique et, d'autre part, d'un réseau d'entreprises offrant des fonctionnalités complémentaires et permettant à Dokeos de maîtriser ses coûts en rétrécissant le socle fonctionnel de son application. L'offre évolue par la suite avec l'offre de service d'accompagnement de l'elearning sur base du SaaS dont l'objectif est de fidéliser les clients, donc sécuriser le flux de revenus récurrents (abonnements), en plus d'apporter un flux complémentaire (cf. Tableau 4).

La stratégie adoptée par Dokeos suit donc un cycle compatible avec la typologie de modèles d'affaires décrite par Jullien et Viseur (2021) et leur évolution en même temps que le cycle de vie du projet (Figure 2). Dokeos se créée sur un modèle d'affaires de personnalisation, suite à des demandes de clients potentiels pour installer et adapter le logiciel, avant d'évoluer rapidement vers un modèle d'édition open source se terminant par le passage à un modèle de double licence (dual licensing). Le logiciel est ensuite réécrit, centré sur un ensemble restreint de fonctionnalités, lors du passage au modèle d'affaires de servitisation. Confirmant l'analyse de Jullien et Viseur (2021) d'une moindre importance du projet libre en phase 3, le modèle de développement open source est purement et simplement abandonné.

Figure 2 : innovation de modèle d'affaires chez Dokeos (inspiré de Jullien et Viseur, 2021)



Ces innovations de modèle d'affaires peuvent être représentées en utilisant la grille d'analyse proposée par Laudien et al. (2017) (cf. Tableau 5).

Tableau 5 : cycle d'innovation de modèle d'affaires de Dokeos (inspiré de Laudien et al., 2017)

	Prestation de services	**SaaS**
Phase de préformation :	Création d'un LMS *open source* en tant qu'utilisateur VH.	Développement de la diffusion par la proposition d'un LMS hébergé.
Barrière de sélection :	Respect de l'idéologie du logiciel libre.	Pas de *vendor lockin*, car *open source*.
Phase 1 (invention) :	Mise en place de la communauté et émergence des demande de services.	ASP (pré-SaaS)
Barrière de mise en œuvre :	Contrôle de la marque par le sponsor (UCL).	Réécriture du logiciel et capacités de gestion d'infrastructure (donc extension / ré-allocation de ressources).
Phase 2 (innovation) :	Création d'un prestataire de services	Offre SaaS (sur base d'un logiciel

	Prestation de services	**SaaS**
	(Dokeos).	interne).
Phase 3 (ajustement) :	Développement de la propriété intellectuelle (double licence avec modules propriétaires).	Offre SaaS (sur base d'un logiciel interne) assortie d'une offre de support (conseil en *elearning*)
Barrière de la nécessité :	Mécontentement de la communauté.	Concurrence de startups en *elearning* bien capitalisées.
Phase 4 (retrait) :	*Fork* du projet et départ de la communauté vers Chamilo.	?

Deux trajectoires peuvent être distinguées : la première sur l'activité de prestation de services (basée sur le projet libre) et la seconde sur l'activité SaaS (après abandon du projet libre). Cette seconde trajectoire n'a pas encore atteint la quatrième phase.

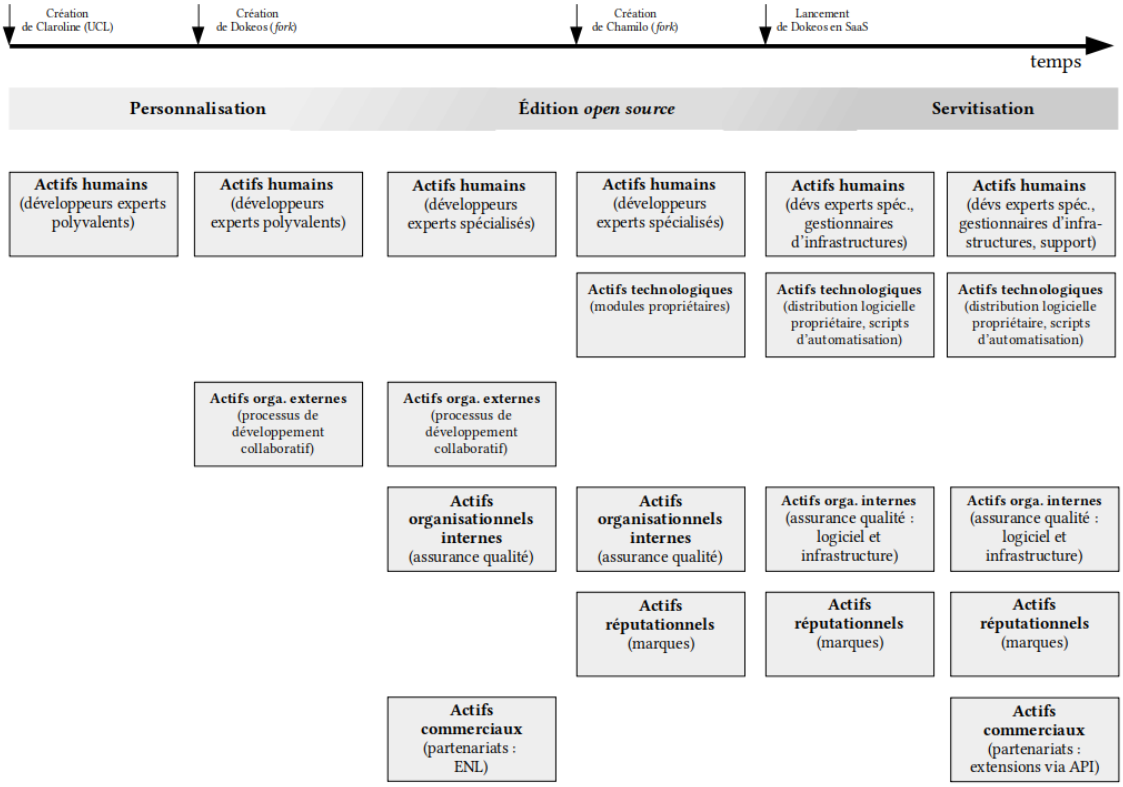
6. Discussion

Dans cette section, nous discuterons la communauté en tant que ressource et la construction des actifs spécifiques chez Dokeos.

6.1. Construction des actifs spécifiques

L'analyse de l'évolution de l'entreprise Dokeos permet de comprendre la construction des actifs spécifiques par l'entreprise (cf. Figure 3).

Figure 3 : construction progressive des actifs spécifiques



Dokeos démarre donc avec un modèle de personnalisation (Jullien et Viseur, 2021). Celui-ci valorise les développeurs polyvalents (actif humain), capables de répondre à la fois au développement de la technologie et aux adaptations demandées par les clients. Elle bénéficie d'un apport de ressources complémentaires par la communauté, ce qui suppose d'organiser la plate-forme de collaboration (actif organisationnel externe). Dokeos évolue ensuite rapidement vers un modèle d'édition open source (cf. Figure 3), ce qui s'accompagne d'une formalisation de la collaboration avec une ENL (entreprise du numérique libre) partenaire (actif commercial) ainsi que d'un renforcement des procédures d'assurance qualité (actif organisationnel interne). Ce modèle est renforcé par le passage à la double licence associé à des extensions propriétaires (actifs technologiques). Par contre, cela se traduit par un départ de la communauté (fork Chamilo) derrière le principale partenaire (destruction d'un actif commercial). Dokeos adopte ensuite un modèle de servitisation et abandonne totalement le modèle de développement open source. Le logiciel est réécrit avec l'aide d'un prestataire spécialisé dans le cloud computing. Ce dernier permet le renforcement d'un actif organisationnel interne (processus de développement). L'entreprise possède la pleine propriété du logiciel (actif technologique). Elle doit recruter du personnel (actif humain) aguerri à la gestion d'infrastructure, laquelle nécessite une extension de l'assurance qualité (actif organisationnel interne, actif technologique). À cette occasion, l'entreprise développe les partenariats (actif commercial) avec des prestataires offrant des fonctionnalités complémentaires au travers de leurs logiciels intégrés par le biais de l'API du SaaS Dokeos. Le logiciel SaaS est ensuite complété par une offre de services dédiés à l'accompagnement en elearning, ce qui suppose de renforcer le personnel (actif humain).

6.2. Rôle de la communauté

La perte de la communauté est-elle toujours une perte pour le producteur du logiciel ? A priori, on pourrait le supposer car elle représente au mieux une perte de ressources, au pire une nouvelle concurrence pour l'éditeur *open source*. En pratique, la réponse est cependant plus nuancée.

Pour Claroline, le départ progressif de la communauté est effectivement une perte (Viseur et Charleux, 2019). La communauté peut assurer différentes tâches : production du logiciel, expérimentation de pistes techniques, création de fonctionnalités expérimentales, développements des usages et expression de besoins métiers… Les institutions dominantes, puis le consortium, se sont recentrées sur les partenaires importants, négligeant les apports liés à l'exploration (technologies, fonctionnalités, usages…) et se détachant progressivement des utilisateurs. Le logiciel perd en notoriété, en contributions, en soutien communautaire aux utilisateurs… Du point de vue des institutions partenaires, il s'agit d'une réelle perte. En effet, dès lors que la pédagogie est perçue comme un élément de différenciation dans la concurrence entre universités, le contrôle d'un projet libre d'enseignement à distance et la disponibilité d'une communauté active, notamment sur les usages, fournit un atout important en matière de transformation pédagogique (p. ex. outillage de nouveaux scénarios pédagogiques[4]) et de résilience dans un contexte pandémique (p. ex. passage en distanciel). Du point de vue du projet, la perte des utilisateurs actifs déforce le projet vis-à-vis des décideurs, susceptibles de décider une migration d'autant plus facilement que la solution est peu ou prou défendue en interne.

Pour Dokeos, la situation est différente (cf. Teece, 2017 puis Laudien et al., 2017, pour la réflexion sur les cycles de vie ; cf. Tableau 5 & Figure 1). Après une naissance du projet au sein de l'UCL (Claroline), l'utilisateur de pointe devient innovateur-entrepreneur et le logiciel trouve son expansion avec la communauté, gagne en notoriété (cf. Google Trends ; cf. Figure 2). Dokeos adapte progressivement son application pour le besoin de ses clients « entreprises » (nouveau segment de clientèle), innove (p. ex. fonctionnalités « *analytics* » dédiées à la mesure de la performance des apprenants) et décide de développer de la propriété intellectuelle sur le projet, dès lors s'éloigne de la communauté. En pratique, Dokeos se concentre sur la relation aux clients tandis que son partenaire Beeznest prend *de facto* en charge la relation à la communauté. Le *fork* s'inscrit dès lors dans la continuité de cette évolution. Dokeos se renouvelle ensuite avec le passage au SaaS, qui voit le développement d'une nouvelle base logicielle. Au vu des exigences de qualité accrues et de la couverture fonctionnelle atteinte, la communauté devient moins indispensable, ce qui amène ici la société à reprendre progressivement en main la solution (privatisation) avec l'aide de partenaires techniques spécialisés (Belighted).

Pour Chamilo, le projet libre est à la fois une finalité et un moyen. La gouvernance est clarifiée avec l'existence, d'une part d'une association (Chamilo) et, d'autre part, d'une

4 À ce niveau, il n'est pas inutile de préciser que Claroline se développe au sein de l'UCL dans un contexte de large réflexion sur l'innovation pédagogique (p. ex. Candi 2000) tandis que l'investissement ultérieur de l'ULyon dans le consortium s'accompagne du développement d'une infrastructure ambitieuse d'enseignement à distance incluant la diffusion de vidéos de cours, et ce, bien avant que la pandémie de la COVID-19 ne systématise de manière contraignante ce genre de pratique.

entreprise (Beeznest). Le positionnement de Beeznest reprend le positionnement de la société Dokeos à ses débuts (cf. Tableau 3) mais en acceptant une logique communautaire plutôt que commerciale pour sa gouvernance (Viseur et Charleux, 2019). La composition de l'ASBL Chamilo est sensiblement différente de celle de l'actionnariat de la société de Yannick Warnier à l'origine du *fork*, même si ce dernier contrôle indirectement le projet par le déploiement de ressources (Schaarschmidt et al., 2015), avec au moins deux développeurs actifs sur le projet, et la Présidence de l'association[5].

Le rôle de la communauté peut ainsi évoluer au fil du temps. Son adhésion n'est par ailleurs pas une fin en soi.

Soit le prestataire adhère à l'*open source* par idéologie, ou plutôt soutient des positions liées au logiciel libre (FSF). Le logiciel doit être libre. Le prestataire est un producteur, plutôt qu'un éditeur, et se rémunère par la prestation de services, sans ajout de « viscosité », sans privatisation.

Soit le prestataire adhère à l'*open source* par pragmatisme. La communauté est une ressource. Au début de la vie du projet, l'entreprise, surtout si elle est peu capitalisée et peut donc difficilement investir sur le projet sans revenus, a besoin de la communauté, pour apporter de nouvelles fonctionnalités, mais aussi pour explorer de nouveaux usages, de nouvelles technologies... Par la suite, la relation à la communauté évolue car l'entreprise dispose d'un socle fonctionnel suffisant (cf. Osterloh & Rota, 2007, qui explique que l'entreprise ouvre classiquement sa propriété intellectuelle en situation de forte incertitude technologique avant de la refermer progressivement) tandis que les exigences de qualité s'accroissent. Il faut donc organiser cette nouvelle relation : soit en renonçant à la communauté, soit en lui réservant un rôle différent, typiquement celui d'exploration (ambidextrie). Teigland et al. (2014), avec leur étude de cas eZ Publish, décrivent ainsi la montée en qualité des équipes internes et, donc, l'utilisation de la communauté pour explorer de nouveaux modules qui, lorsqu'ils sont intéressants, sont ensuite redéveloppés dans les règles de l'art.

Quel rôle pourrait-on dès lors accorder à la communauté associée au projet libre au fur et à mesure que l'entreprise focale avance dans le cycle de vie de son modèle d'affaires ?

Jullien et Zimmermann (2011) proposent une typologie d'utilisateurs : les utilisateurs VH (Von Hippel, cf. Von Hippel, 2006), les utilisateurs KM (Kogut Meitu, cf. Kogut & Metiu, 2001) et les utilisateurs naïfs. En réalité, les apports des utilisateurs diffèrent suivant les profils : tests de technologies, modules expérimentaux, expression des besoins, publicité pour le projet…, ce qui implique, si l'on souhaite conserver la communauté en tant que ressource, au fil du cycle de vie du prestataire, d'adopter une gestion évolutive de la communauté. Cette dernière ne va cependant pas de soi et s'accompagne régulièrement de conflits pouvant aller jusqu'à la rupture (*fork*). Si le projet libre fournit un terrain (la communauté) intermédiaire entre la firme et le marché, et si la communauté fournit un support à la création de valeur, elle augmente aussi les coûts de transaction en imposant une négociation des changements (Ghertman, 2006). La communauté constitue un mode de gouvernance, au sens de Williamson, intermédiaire entre la firme et le marché. Si les membres de la communauté sont autonomes et ne sont donc pas supposés se plier aux décisions des responsables du projet libre, il n'en reste pas moins que les licences libres et *open source* (contrats) fluidifient les transactions au sein de la communauté en homogénéisant les droits et les devoirs des différentes parties-prenantes. Ces coûts peuvent aussi augmenter suite aux tentatives de

5 Cf. https://chamilo.org/fr/chamilo/.

contrôle exercées, par exemple, par d'autres prestataires déployant des ressources (Schaarschmidt et al., 2015).

Tableau 6 : création et captation de valeur (Dokeos)

	Value-in-use	**Value-in-exchange**
Value creation	*Value realization* Aucune (reversement à la communauté), puis création de modules spécifiques (2010) puis réécriture complète du logiciel (pleine propriété) en vue de sa SaaSification (2020).	*Value provision* Développement du projet initialement sous licence *open source* à la suite du *fork* de Claroline.
Value capture	*Value partake* Prestation de services sur base de Dokeos avec une situation de coopétition avec les partenaires étrangers et les concurrents.	*Value negociation* Lancement d'une offre sous double licence avec une discussion sur l'IP des modules privatisés (*fairness*, réversion).

L'innovation de modèle d'affaires se révèle aussi être un vecteur de conflits. Ces derniers surviennent typiquement autour du contrôle des ressources (p. ex. marques), de la discussion des objectifs (liés notamment aux segments de clientèle privilégiés) et des modalités de captation de valeur (cf. Chesbrough et al., 2018 et Tableau 6).

La communauté associée à un projet libre n'apparaît par ailleurs pas comme l'instrument de l'éditeur open source. Dans l'écosystème élargi étudié dans cet article, la communauté ne meurt pas lorsque l'éditeur choisit de s'en passer : elle va là où la gouvernance effective lui est le plus favorable et permet de contribuer au projet. Après avoir suivi en confiance le leader du projet Claroline dans son projet Dokeos, elle rejoint ainsi le projet Chamilo. La gouvernance de ce dernier, conforme à la logique communautaire (Viseur et Charleux, 2019), augure une stabilisation de la communauté autour de cet énième projet libre.

7. Conclusion

Dans cette recherche, nous avons analysé l'évolution des modèles d'affaires d'un prestataire open source (Dokeos). Ce cas nous a permis d'améliorer notre compréhension du rôle de la communauté du point de vue d'un prestataire open source ainsi que la dynamique de construction d'actifs spécifiques permettant de sécuriser le modèle d'affaires et protéger l'entreprise de l'action de la concurrence. La communauté peut ainsi apparaître, soit comme une fin en soit, dès lors que l'éditeur adhère aux valeurs du logiciel libre, soit comme une ressource que l'on mobilise en fonction des besoins de l'entreprise (contributions, traductions, tests, promotion, self support…), qui peut dès lors être pragmatiquement entretenue, ré-allouée ou abandonnée. Dès lors que le prestataire renonce à la communauté en tant que ressource, il peut se décharger de son organisation (actif organisationnel externe) mais doit compenser cette perte par le renforcement de son personnel spécialisé (actif humain) et par les partenariats clefs (actif commercial). Ces résultats illustrent par ailleurs l'argument de Chesbrough et al. (2018) sur l'importance de bien gérer la déception et l'équité au sein d'un écosystème d'innovation ouverte.

8. Références

Adner, R., & Kapoor, R. (2010). Value creation in innovation ecosystems: How the structure of technological interdependence affects firm performance in new technology generations. Strategic management journal, 31(3), pp. 306-333.

Aleks, R., Maffie, M. D., & Saksida, T. (2020). The role of collective bargaining in a digitized workplace. *Reimagining the Governance of Work and Employment*, *85*.

Alier, M., Casañ, M. J., & Piguillem, J. (2010). Moodle 2.0: Shifting from a learning toolkit to a open learning platform. In *International Conference on Technology Enhanced Learning* (pp. 1-10). Springer, Berlin, Heidelberg.

Amit, R., & Zott, C. (2001). Value creation in e-business. Strategic management journal, 22(6□7), pp. 493-520.

Bajoit, G. (1988). Exit, voice, loyalty... and apathy. Les réactions individuelles au mécontentement. *Revue française de sociologie*, 1988, 29-2. pp. 325-345.

Baldwin, C., Hienerth, C., & Von Hippel, E. (2006). How user innovations become commercial products: A theoretical investigation and case study. *Research policy*, *35*(9), pp. 1291-1313.

Bidan, M., Biot-Paquerot, G., Chaboud, M. C., & Lentz, F. M. (2020). Inversion du domaine de l'adoption: les technologies latentes. *Management & Datascience*, *4*(2).

Chesbrough, H. W. (2003). *Open innovation: The new imperative for creating and profiting from technology*. Harvard Business Press.

Chesbrough, H., Lettl, C., & Ritter, T. (2018). Value creation and value capture in open innovation. *Journal of Product Innovation Management*, *35*(6), pp. 930-938.

Charleux, A., Mione, A. (2018). Les business models de l'édition open source ; le cas des logiciels. *Finance Contrôle Stratégie*, (NS-1).

Demil, B. et Lecocq, X. (2003). Comment exploiter brevets et marques. L'expansion management review, pp. 88-95.

Dierickx, I., & Cool, K. (1989). Asset stock accumulation and sustainability of competitive advantage. Management science, 35(12), pp. 1504-1511.

Di Penta, M., German, D. M., Guéhéneuc, Y. G., & Antoniol, G. (2010, May). An exploratory study of the evolution of software licensing. In *2010 ACM/IEEE 32nd International Conference on Software Engineering* (Vol. 1, pp. 145-154). IEEE.

Farrell, D. (1983). Exit, voice, loyalty, and neglect as responses to job dissatisfaction: A multi--dimensional scaling study. Academy of Management Journal, 26(4), 596-607.

Fautrero, V., & Gueguen, G. (2012). Quand la domination du leader contribue au déclin. Revue française de gestion, (3), pp. 107-121.

Filippova, A., Cho, H. (2016, February). The effects and antecedents of conflict in free and open source software development. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing* (pp. 705-716).

Fitzgerald, B. (2006). The transformation of open source software. *MIS quarterly*, 587-598.

Foss, N. J., & Saebi, T. (2017). Fifteen years of research on business model innovation: How far have we come, and where should we go?. *Journal of Management*, *43*(1), 200-227.

Gadray, J. (2003). *Socio-économie des services*. La Découverte. Paris.

Ghertman, M. (2006), Oliver Williamson et la théorie des coûts de transaction. Revue française de gestion, (1), pp. 191-213.

Grima, F., & Glaymann, D. (2012). A revisited analysis of the exit-voice-loyalty-neglect model: contributions of a longitudinal and conceptually extended approach. *M@ n@ gement*, 1-41.

Hirschman, A. O. (2017). *Exit, voice, loyalty. Defection et prise de parole*. Éditions de l'Université libre de Bruxelles.

Jensen, C. et Scacchi, W. (2005). Collaboration, leadership, control, and conflict negotiation and the netbeans. org open source software development community. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences* (pp. 196b-196b). IEEE.

Jullien, N. et Zimmermann, J.-B. (2011). Floss firms, users and communities : a viable match? *Journal of Innovation Economics*, 1(7), pp. 31-53.

Kogut, B. M., et Metiu, A. (2001). Open source software development and distributed innovation. *Oxford Review of Economic Policy*, 17(2), pp. 248-264 .

Laudien, S. M., & Daxböck, B. (2017). Understanding the lifecycle of service firm business models: a qualitative□empirical analysis. *R&D Management*, *47*(3), pp. 473-483.

Lee, S., Baek, H., & Jahng, J. (2017). Governance strategies for open collaboration: Focusing on resource allocation in open source software development organizations. International Journal of Information Management, 37(5), pp. 431-437.

Lepak, D. P., Smith, K. G., & Taylor, M. S. (2007). Value creation and value capture: A multilevel perspective. Academy of management review, 32(1), pp. 180-194.

March, J. G. (1991). Exploration and exploitation in organizational learning. Organization science, 2(1), pp. 71-87.

Massa, L., Tucci, C. L., & Afuah, A. (2017). A critical assessment of business model research. Academy of Management Annals, 11(1), pp. 73-104.

O'mahony, S., & Ferraro, F. (2007). The emergence of governance in an open source community. *Academy of Management Journal*, *50*(5), 1079-1106.

Onetti, A., & Verma, S. (2009). Open source licensing and business models. *IUP Journal of Knowledge Management*, *7*(1), 68.

Osterloh, M. et Rota, S. (2007). Open source software development. just another case of collective invention? Research Policy, 36(2), pp. 157-171.

Osterwalder, A., & Pigneur, Y. (2011). *Business Model nouvelle génération: Un guide pour visionnaires, révolutionnaires et challengers* (Vol. 1). Pearson.

Rusbult, C., Zembrodt, I., & Gunn, L. (1982). Exit, voice, loyalty, and neglect: Responses to dissatisfaction in romantic involvements. Journal of Personality and Social Psychology, 43(6), 1230-1242.

Shahrivar, S., Elahi, S., Hassanzadeh, A., & Montazer, G. (2018). A business model for commercial open source software: A systematic literature review. *Information and Software Technology*, *103*, pp. 202-214.

Teece, D. J. (2018). Business models and dynamic capabilities. Long range planning, 51(1), pp. 40-49.

Teigland, R., Di Gangi, P. M., Flåten, B. T., Giovacchini, E., & Pastorino, N. (2014) Balancing on a tightrope: Managing the boundaries of a firm-sponsored OSS community and its impact on innovation and absorptive capacity. Information and Organization, 24(1), pp. 25-47.

Trott, P. (2012), *Innovation management and new product development*, Prentice Hall.

van Wendel de Joode, R. (2004). Managing conflicts in open source communities. *Electronic Markets*, *14*(2), 104-113.

Viseur, R. (2012). Forks impacts and motivations in free and open source projects. *International Journal of Advanced Computer Science and Applications*, *3*(2), 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. (2013b). Identifying success factors for the mozilla project. In *IFIP International Conference on Open Source Systems* (pp. 45-60). Springer, Berlin, Heidelberg.

Viseur, R. (2007). Gestion de communautés Open Source. In 12ème Conférence de l'Association Information et Management, Lausanne (Suisse).

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, Cham.

Viseur, R. Charleux, A. (2019). Changement de gouvernance et communautés open source : le cas du logiciel Claroline. Innovations, (1), pp. 71-104.

Viseur, R. & Charleux, A. (2021), Open source communities and forks: a rereading in the light of Albert Hirschman's writings, Open Source Systems 2021.

Von Hippel, E. (2001). Learning from open-source software. *MIT Sloan management review*, *42*(4), pp. 82-86.

Von Hippel, E. (2006). *Democratizing innovation* (p. 216). the MIT Press.