



HAL
open science

Open Source Communities and Forks: A Rereading in the Light of Albert Hirschman's Writings

Robert Viseur, Amel Charleux

► To cite this version:

Robert Viseur, Amel Charleux. Open Source Communities and Forks: A Rereading in the Light of Albert Hirschman's Writings. 17th IFIP International Conference on Open Source Systems (OSS), May 2021, Lathi/virtual event, Finland. pp.59-67, 10.1007/978-3-030-75251-4_6 . hal-03254062

HAL Id: hal-03254062

<https://inria.hal.science/hal-03254062v1>

Submitted on 8 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Open source communities and forks: a rereading in the light of Albert Hirschman's writings

Robert Viseur¹ and Amel Charleux²

¹ University of Mons

² University of Montpellier

robert.viseur@umons.ac.be

amel.charleux@umontpellier.fr

Abstract. The literature dedicated to free and open source software emphasizes the support given by the community to software producers. However, the community is also a place of conflict and can sometimes experience violent splits (forks). Communities can show different forms of resistance to change. In this research, we propose a re-reading of these mechanisms of opposition in light of Albert Hirschman's theory (exit, voice, loyalty). We present the fork as a new form of defection (exit) allowed by licenses and discuss the rationality of choice for the economic actors who implement it.

Keywords: open source, business model innovation, governance, fork.

Introduction

The literature on free and open source software emphasises the positive role of the community in the efforts of the software producer to ensure its development and popularity (Shahrivar & al., 2018). However, the literature points out the possibility of conflicts that could lead to a split in the community and, therefore, to the creation of a competing project (fork) based on the source code of the original project (Viseur, 2012; Viseur & Charleux, 2019). In a recent study dedicated to Claroline software, Dokeos (fork of Claroline) and Chamilo (fork of Dokeos), Charleux & al. (2019), then Viseur and Charleux (2019), note that the community is also a force of opposition resisting changes initiated by the producer in a context of business model innovation. The opposition mechanisms identified present surprising similarities with the alternative actions identified by Hirschman (2017) concerning the consumer (or citizen) faced with a declining organisation. This article therefore proposes a re-reading of the work of Charleux and his co-authors (2019) with regard to Albert Hirschman's theory.

1. Community as a source of value

The term "free software" has been defined by the Free Software Foundation (FSF) through 4 freedoms: freedom of use, freedom of study, freedom of distribution and

freedom of redistribution. The term "open source" was subsequently defined by the Open Source Initiative (OSI) on the basis of 10 criteria including in particular freedom of redistribution, access to the source code, creation of derivative works and respect for authorship. Beyond the difference in terminology, where the FSF sees free software as a political project oriented towards sharing and user emancipation, the OSI puts forward the cooperative development model as well as the associated business models and licenses (Benkeltoum, 2011).

Free / open source software can arise in a variety of contexts. Firstly, it can be created by one (or more) user(s) concerned with solving a problem they encounter, in accordance with Raymond's (1999) quote: "*Every good work of software starts by scratching a developer's personal itch*". This creation will typically take place in a professional context, as shown by the examples of Apache (Franke & Von Hippel, 2003) and Claroline (Viseur & Charleux, 2019). Secondly, it may be created by companies in order to pool resources and promote the dissemination of a technology or standard (Adatto, 2013). Thirdly, it can be produced by a company in an entrepreneurial context with the aim of subsequently meeting customer needs. This type of open source producer is paid for by services and sometimes licences (e.g. dual licensing; cf. Välimäki, 2003; Charleux & Mione, 2018) while relying on a community to develop the software and disseminate the brand (Fitzgerald, 2006). Examples already studied include eZ Publish (Teigland & al., 2014) and MySQL (Välimäki, 2003).

2. The community as a brake

The community is therefore considered, for open source companies, as an important resource and a key factor in its success (Shahrivar & al., 2018). Thanks to it, the open source publisher would benefit from a reduced development cost because, on the one hand, volunteer developers would code for free and, on the other hand, users would report problems in the software (Shahrivar & al., 2018). However, the community can also be a source of disillusionment and difficulties. The commitment of developers and users is not guaranteed, either in quantity or quality. Viseur (2007) thus reports, based on 6 case studies (eZ Publish, Claroline, Exo, Plume CMS, Ekiga and Jext), that "*the most frequent contributions concern bug reports, translations and, more rarely, the addition of new functionalities*". The hope of seeing developers coding for free is therefore put into perspective by open source project managers. Teigland & al (2014) reveal the gap between the quality requirements of a publisher (eZ Systems) and the contributions in source code brought by the community (often in the form of extensions to the eZ Publish project). Viseur and Charleux (2019) make the same observation in the case of the Claroline project. The authors also highlight the difficult animation of the community and analyse two concrete cases of community splits (forks) (Dokeos and Chamilo). The community can therefore be a support (contributions, feedback of errors...), but also sometimes a force of resistance that can lead to new forms of competition through the forks.

The forks are motivated by several phenomena. Viseur (2012), in his analysis of the forks of 26 popular projects, isolates six motivations for forking a project: stop-

ping the original project, technical motivations, license changes, conflict over brand ownership, problems of project governance, cultural differences and the search for new directions of innovation. Changes in business models not negotiated with the community can also lead to resistance and defection through the creation of forks (Charleux & al. 2019). The negotiation of strategic parameters such as governance appears in this context to be essential to maintain community buy-in and investment (Viseur & Charleux, 2019). Alignment between the interests of project promoters and their community must be preserved to guarantee the long-term success of projects: “*effective governance and work practices that are appreciated by community members is fundamental for long-term sustainability*” (Gamalielsson & Lundell, 2014). Alignment of strategy, business model and governance emerge as a necessary and difficult balance to achieve (Viseur & Charleux, 2019). Markus (2007) defines open source governance as the set of means implemented for the guidance, control and coordination of fully or partially autonomous organisations and individuals on behalf of an open source development project to which they collectively contribute. It combines it with a set of characteristics including the ownership of assets (such as trademarks, licenses and copyrights), the objectives of the project, conflict resolution and rule change, and the modalities of access to tools. Viseur (2012) shows that these elements (via diverging technical choices, conflicts over brand ownership, changes in licenses, etc.) emerge as major elements of conflicts within communities that can lead to forking. The case of license changes is emblematic of these tensions that can arise between the producer and his community (Viseur & Robles, 2015). License changes take place in very distinct contexts, particularly in relation to the need to adapt to the environment or change the business model. The development of cloud computing over the last ten years or so has shown how a company can move from a service delivery model to a publishing model and then to a service operator (SaaS) model by adapting the terms of its license (Viseur, 2013). These changes in project parameters may alter the conditions for value creation and appropriation within the project community (Charleux & Mione, 2018). If, for some, the changes can be positive and represent opportunities, for others, these changes can be harmful, leading to opposition and conflicts that can go as far as the fork.

Table 1. Gradation of opposition mechanisms (adapted from Charleux & al., 2019).

Public expres- sion of discontent.	Use of software associated with a cessation of contri- butions.	Stopping the use of the software and migration.	Fork.
---------------------------------------	--	---	-------

Conflicts within communities, however, do not always lead to forks and do not always manifest the same violence. The expression of discontent can be gradual (Charleux & al., 2019). Thus, in the particular situation of a strategic change in business model, the community may (1) publicly express its dissatisfaction, (2) continue to use the software but stop contributions, (3) stop using the software and/or (4) fork (see Table 1). The issue of negotiation with the community in the specific context of a

business model innovation (BMI) highlights the inertia brought about by the community in the face of a situation of change. This type of situation is particularly in line with the issues of equity and reversion raised in the field of open innovation by Chesbrough, Lettl and Ritter (2018). The producer has to deal with community values and disappointment with new policies.

3. Exit and voice (Hirschman)

In an early book originally published in 1970 (*Exit, Voice and Loyalty. Response to Decline in Firms, Organisations and States*), Albert Hirschman (2017) asks the questions (1) of the actions of consumers who are dissatisfied with a product or service and (2) of the means available to businesses to remedy their decline. Hirschman identifies three mechanisms used by consumers. The first is defection (exit): faced with deterioration in quality, the consumer brings competition into play. The second is voice: consumers express their dissatisfaction. The third is loyalty: the consumer resigns himself to defects through inertia, loyalty or lack of a real alternative.

Hirschman evaluates the effectiveness of these different reactions and notes in particular that defection is more or less effective in a competitive environment due to the inefficient chasing of customers. He identifies an optimum consisting of a balance between passive customers leading to the purchase of the product and vigilant customers serving as a warning signal for the company. He thus imagines that a monopoly based on the search for profits can be more effective when the speaking out allows the start of a recovery movement. As for the sacrifice of remaining loyal, it can be explained by the will to exert influence, the expectation of results (if loyalty is combined with a collective complaint), the costs of change and loyalty (judged not to be fully rational as opposed to other actions). Furthermore, Hirschman attempts in his analysis to mix economics and politics. He thus extends his analysis to non-profit organisations (e.g. political parties) and states, showing that the credibility of a threat of defection coupled with voice justifies loyalty because it gives hope for a turnaround in the organisation.

To the three mechanisms of discontent identified by Hirschman (2017), Bajoit (1988) proposes a fourth: apathy. In this schema, the dissatisfied individual can either leave (exit) or stay. If he stays, he can either protest (voice) or remain. In the latter case, he can still participate actively (loyalty) or passively (apathy). Apathy is characterised by resignation and is a form of mistrust. It does not lead to conflict and maintains social control. It plays a moderating role in the mechanisms described by Hirschman by preventing the collapse of the organisation following the flight of its members.

4. Fork as a new form of exit

4.1. Opposition mechanisms and Hirschman's model

In his work, Hirschman attempts to reconcile the economic (favouring exit) and political (favouring voice) spheres. However, free and open source software covers both spheres. Free software, which appeared in the 1980s, develops a political and ethical project of user emancipation, whereas open source places greater emphasis on the industrial and economic dimensions (Benkeltoum, 2011; Fitzgerald, 2006; Charleux & Mione, 2018). These two conceptions coexist within communities. Moreover, the management of an open source project has in practice a political side (governance) and an economic side (business model), while the rationality of the actors is complemented by a strong moral dimension (cf. Hirschman's normative utilitarianism). Charleux, Viseur and Mione (2019) identify several gradations among the mechanisms of community resistance (cf. Table 1). Compared to Hirschman (cf. Figure 1), the first (public expression of discontent) corresponds to a form of voice, while the third (migration) and the fourth (fork) constitute a form of exit. As for passive use, it is a form of loyalty but can be compared to apathy (Bajoit, 1988), allowing a reasonably stable user base to be maintained. In practice, the community provides the producer with continuous feedback on its choices, and also elements likely to influence them (if the producer listens to them!), sometimes in opposition to the opinion of internal teams (Teigland & al., 2018).

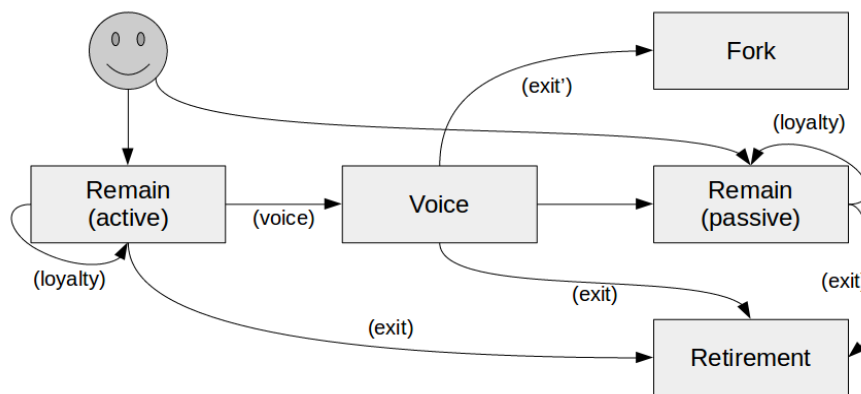


Fig. 1. Opposition mechanisms.

4.2. Extension to open hardware

Fauchart et al. (2017) provide material for understanding these opposition mechanisms in open hardware. The authors have mainly studied Makerbot, a company active in desktop 3D printers. Its products were initially developed in an open manner, before the development process (“*open realease*’ but ‘*closed development*’”) and

then the products themselves were gradually closed, a relatively common strategy when technological uncertainty tends to be reduced and the innovative company seeks to protect itself from possible imitators (Fauchart & al., 2017; Osterloh & Rota, 2007). The expression of discontent has developed as a result of various events: the gradual closure of the project, a fundraising campaign, the discovery of the conditions of use relating to intellectual property on the Thingiverse platform (also launched by Makerbot) and the filing of patents. The community has therefore reacted to various forms of misappropriations that do not necessarily violate the project's license but are at odds with the commonly accepted culture and norms (if not explicit). The misappropriations took different forms: posting on influential blogs (voice), stopping contributions (apathy), refusing to buy the product again (exit), criticising the brand (voice) and calling for a boycott (i.e. a combination of voice and exit). The company experienced a significant commercial decline as a result of these events, but also the deterioration in the quality of the machines.

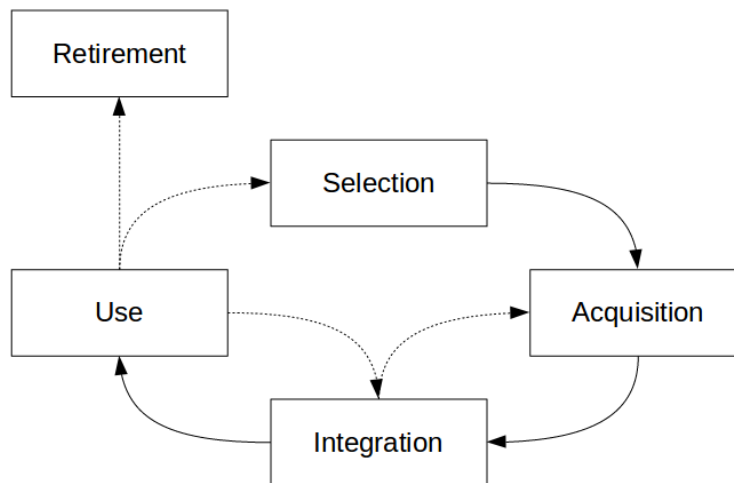


Fig. 2. The elements of Total Cost of Ownership (TCO).

4.3. Justification of apathy

The intangible nature of the software should not lead one to believe that the cost of change is systematically low. Shaikh and Cornford (2011) thus show that the Total Cost of Ownership of software relates to a set of operations (cf. Figure 2) including the exploration of possible alternatives, the acquisition of the chosen solution, its integration into work procedures, its use and, if the decision is taken to no longer capitalise on the same software, the exit from the solution (with a view to migration to another solution). On the one hand, the user of software within an organisation does not generally have the freedom to choose the software he uses. On the other hand, the company itself has limited degree of freedom. Thus, exploration does not require development skills, but rather the ability to define needs and evaluate offers, i.e. skills that lay users do not have (Kogut & Metiu, 2001). Moreover, companies generally

have to deal with legacy systems and technological integration efforts, which can be likened to a form of path dependency. The desire to migrate is curbed by the increasing returns on adoption (Foray, 2002) as well as by vendor lock-in processes (Zhu & Zhou, 2012) which increase the cost of defection (migration). Apathy is therefore hardly surprising even if history also contains rare examples to the contrary (e.g. PHP Nuke and its numerous forks).

4.4. Rationality of a fork

The fork appears to be a widespread source of fear in the open source industry (Viseur, 2012) even if, for others, it emerges as a form of invisible hand contributing to the sustainability of projects (Nyman & al., 2012). Nyman & al. (2012) thus insist on the fact that the very existence of this possibility of a fork stimulates the listening and consideration given by project leaders to community contributors. In this sense, the threat of a fork is seen as a factor of community recovery when its management leaves something to be desired. However, it assumes that the conflict lasts long enough, as Hirschman predicts, to allow for recovery. While this is often the case, whether the fork occurs (e.g. LibreOffice.org) or not (e.g. Java) in the end, the conflict can also too quickly lead to a fork (e.g. Dokeos), without the original project having time to adjust its governance, eventually leading to the death of the original project. While the fork may be initiated, sometimes suddenly, by a single individual (e.g. Dokeos), it is also more often the result of prior negotiation and coordination between influential project members (e.g. Chamilo) (Viseur & Charleux, 2019). From the user's point of view, however, the fork facilitates defection because it lowers migration costs.

In order to better understand the economic rationality behind a fork, we propose to deepen the understanding of forks initiated (or supported) by commercial companies. For the latter, the software can be a key resource when it enables them to gain a competitive advantage on a market (e.g. dissemination of a standard through an open source implementation; Adatto, 2013) or when its rate of evolution is rapid (cumulative aspect). In the latter case, the value created is more a “*value in exchange*” than a “*value in use*” (Chesbrough, Lettl and Ritter, 2018), resulting in a continuous flow of contributions that the company must be able to absorb. The company can then seek to gain control over the project through leadership (e.g., sponsoring or recruiting influential members) or by deploying resources (e.g., development capabilities) (Schaarschmidt & al., 2015). When both of these options fail, the fork can be an effective means of parallel takeover. By creating a project that competes with the initial project, the company can achieve its strategic objectives by benefiting from the impetus of the initial project. This is how Google, through the fork of the WebKit project (rendering engine), itself forked with KHTML by Apple, was able to develop and deploy its own project called Blink, a Chromium component used as a basis for several browsers (including Google Chrome). The fork provided Google with a solid foundation to develop its own rendering engine project. On the one hand, Google wanted to be able to make modifications to WebKit on a larger scale to meet different needs from those of the WebKit project (Baysal & al., 2016). The fork therefore saves on

transaction costs. On the other hand, control of this technology provides the company with a strategic instrument to influence web standards (e.g. HTML5; cf. Fukami, 2016) and more closely control access to the web pages on which its advertising platform business model depends (Osterwalder and Pigneur, 2011; Srnicek, 2018).

Conclusion

This research enabled us to propose a synthesis on the motivations behind the forks. Based on Charleux et al. (2019), exploiting the field offered by the Claroline, Dokeos (fork of Claroline) and Chamilo (fork of Dokeos) projects, we presented the opposition mechanisms mobilised by open source communities. We then showed the similarity between these mechanisms of opposition and the mechanisms of expression of discontent, i.e. exit, voice and loyalty, proposed by Albert Hirschman (1970, 2017) to explain the behaviour of an agent faced with the decline of an organisation. More specifically, we analysed the fork reaction as an additional manifestation to be counted as an action of exit. Drawing on Fauchart et al. (2017), which offers rich material on conflicts in the Makerbot community, we also showed the applicability of this analytical framework to an open hardware project.

This research represents a first step towards understanding open source communities in the light of Albert Hirschman's writings and contributes to an effort to theorise the mechanisms of opposition within communities. Two in-depth studies seem to us particularly worthy of interest. On the one hand, the levels of expertise and commitment of the members of an open source community (see Crowston & Howison, 2005) could be distinguished in order to differentiate the opposition mechanisms implemented and discuss their dangerousness for the short-term stability of the project. On the other hand, these opposition mechanisms could be associated with indicators that can be calculated in an automated manner and thus allow for an anticipation of the most dangerous community reactions (e.g. call for boycott and fork).

Acknowledgements

This research was supported by the European Regional Development Fund as part of the FabricAr3v cross-border project.

References

1. Adatto, T.: Standards ouverts et implémentations FLOSS (Free Libre Open Source Software) : vers un nouveau modèle synergique de standardisation promu par l'industrie du logiciel. *Terminal: Technologie de l'Information, Culture, Société*, (113-114), 137-170 (2013).
2. Bajoit, G.: Exit, voice, loyalty... and apathy. Les réactions individuelles au mécontentement. In: *Revue française de sociologie*, 1988, 29-2. pp. 325-345 (1988).

3. Baysal, O., Kononenko, O., Holmes, R., & Godfrey, M. W.: Investigating technical and non-technical factors influencing modern code review. *Empirical Software Engineering*, 21(3), 932-959 (2016).
4. Benkeltoum, N.: Regards sur les stratégies de détournement dans l'industrie open source. *Vie sciences de l'entreprise*, (1), 72-94 (2011).
5. Charleux, A., Viseur, R., & Mione, A.: Open Source Innovation: Enabler or Hinderer of Business Model Dynamics. *Actes de XXVIIe Conférence Internationale de Management Stratégique (AIMS)* (2019).
6. Chesbrough, H., Lettl, C., & Ritter, T.: Value creation and value capture in open innovation. *Journal of Product Innovation Management*, 35(6), 930-938 (2018).
7. Crowston, K. & Howison, J.: The social structure of Free and Open Source Software development. *First Monday*, 10(2) (2005).
8. Fitzgerald, B.: The transformation of open source software. *MIS quarterly*, 587-598 (2006).
9. Fauchart, E., Rayna, T., Striukova, L.: Is selling caring? Norms regulating commercialisation and sharing behaviour with the open hardware RepRap. *Proceedings of the "XXVI^{ème} Conférence Internationale de Management Stratégique" (AIMS)*, Lyon (France) (2017).
10. Foray, D.: Innovation et concurrence dans les industries de réseau. *Revue française de gestion*, (3), 131-154 (2002).
11. Franke, N., & Von Hippel, E.: Satisfying heterogeneous user needs via innovation toolkits: the case of Apache security software. *Research policy*, 32(7), 1199-1215 (2003).
12. Fukami, Y.: Open architectural competition strategy: Google's approach to innovation through standardization. In *IWSECO@ ICIS* (pp. 80-94) (2016).
13. Gamalielsson, J., & Lundell, B.: Sustainability of Open Source software communities beyond a fork: How and why has the LibreOffice project evolved?. *Journal of Systems and Software*, 89, 128-145 (2014).
14. Hirschman, A. O.: Exit, voice, loyalty. *Defection et prise de parole*. Éditions de l'Université libre de Bruxelles (2017). ISBN : 978-2-8004-1625-0.
15. Hirschman, A. O.: Exit, voice, and loyalty: Responses to decline in firms, organizations, and states. *Harvard University Press* (1970).
16. Kogut, B. M., et Metiu, A.: Open source software development and distributed innovation. *Oxford Review of Economic Policy*, 17(2) : 248-264 (2001).
17. Mäenpää, H., Munezero, M., Fagerholm, F., & Mikkonen, T.: The many hats and the broken binoculars: State of the practice in developer community management. In *Proceedings of the 13th International Symposium on Open Collaboration* (pp. 1-9) (2017).
18. Markus, M. L.: The governance of free/open source software projects: monolithic, multidimensional, or configurational?. *Journal of Management & Governance*, 11(2), 151-163 (2007).
19. Nyman, L., Mikkonen, T., Lindman, J., & Fougère, M.: Perspectives on code forking and sustainability in open source software. In *IFIP International Conference on Open Source Systems* (pp. 274-279). Springer, Berlin, Heidelberg (2012).
20. Osterloh, M., & Rota, S.: Open source software development—Just another case of collective invention?. *Research Policy*, 36(2), 157-171 (2007).
21. Osterwalder, A., & Pigneur, Y.: *Business Model nouvelle génération: Un guide pour visionnaires, révolutionnaires et challengers* (Vol. 1). Pearson (2011).

22. Raymond, E.: The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12(3), 23-49 (1999).
23. Schaarschmidt, M., Walsh, G., & von Kortzfleisch, H. F.: How do firms influence open source software communities? A framework and empirical analysis of different governance modes. *Information and Organization*, 25(2), 99-114 (2015).
24. Shahrivar, S., Elahi, S., Hassanzadeh, A., & Montazer, G.: A business model for commercial open source software: A systematic literature review. *Information and Software Technology*, 103, 202-214 (2018).
25. Shaikh, M., Cornford, T.: Total cost of ownership of open source software : a report for the UK. Cabinet Office supported by OpenForum europe (2011).
26. Srnicek, N.: *Capitalisme de plateforme. L'hégémonie de l'économie numérique*. Lux (2018).
27. Teigland, R., Di Gangi, P. M., Flåten, B. T., Giovacchini, E., & Pastorino, N.: 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), 25-47 (2014).
28. Välimäki, M.: Dual licensing in open source software industry. *Systemes d'Information et Management*, 8(1), 63-75 (2003).
29. Viseur, R., & Charleux, A.: Changement de gouvernance et communautés open source: le cas du logiciel Claroline. *Innovations*, (1), 71-104 (2019).
30. Viseur, R., & Robles, G.: 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 (2015).
31. Viseur, R.: Evolution des stratégies et modèles d'affaires des éditeurs Open Source face au Cloud computing. *Terminal. Technologie de l'information, culture & société*, (113-114), 173-193 (2013).
32. 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.
33. Viseur, R.: Gestion de communautés Open Source. In *12ème Conférence de l'Association Information et Management, Lausanne (Suisse)* (2007).
34. Zhu, K. X., & Zhou, Z. Z.: Research note—Lock-in strategy in software competition: Open-source software vs. proprietary software. *Information Systems Research*, 23(2), 536-545 (2012).