

Etude empirique sur le comportement des développeurs et son impact sur le développement de logiciels open source

Hind Benbya, Nassim Belbaly, Régis Meissonier

Groupe Sup de Co Montpellier
2300 av. des Moulins, Montpellier, France
hbenbya@supco-montpellier.fr
nbelbaly@supco-montpellier.fr
rmeissonier@supco-montpellier.fr

Résumé

Comprendre le comportement des développeurs et son impact sur le développement des logiciels libres fait partie des questions scientifiques en vogue dans la recherche en Systèmes d'Information. Les travaux de recherche les plus saillants sur l'open source ont permis de découvrir les motivations des participants. Pour autant, comprendre les attitudes, les implications et les participations, les antécédents et les effets de leurs comportements, demeurent des interrogations scientifiques retenant une attention particulière. Ceci est à la fois révélateur du côté novateur comme du manque de corpus théorique solide du « phénomène open source ». Dans cet article, qui mobilise des courants issus de la psychologie et de la *social exchange theory*, nous développons un modèle théorique sur les antécédents et les résultats associés aux comportements des développeurs. La partie empirique permettant de tester ledit modèle est effectuée par l'intermédiaire d'une enquête en ligne administrée auprès de SourceForge.net, la plus importante plateforme de développement de projets open source.

Mots clés : Logiciels Open Source, Comportement des développeurs, *Social exchange theory*, Psychologie.

1 Introduction

Les logiciels open source sont de plus en plus considérés comme des alternatives technologiques viables aux logiciels propriétaires. Alors que les fondements du mouvement open source reposent sur des conditions autres que la rentabilité économique, de nombreuses entreprises (dont parmi les plus célèbres : IBM, Sun Microsystems ou Oracle) de même que des établissements scientifiques tels la Nasa sponsorisent des projets de logiciels libres (comme OSSP—IBM Open Source Zone¹, SunSource.net², Oracle Free and Open Source Software³, NASA Open Source Software⁴). Ce principe remet donc en question les théories traditionnelles selon lesquelles les comportements des agents économiques sont basés sur des logiques de coûts et de bénéfices [Bonaccorsi & Rossi, 03]. Pour comprendre le « phénomène open source », de nombreuses études ont été conduites sur les motivations expliquant l'implication bénévole de développeurs à la création de logiciels. Les résultats de ces études ont permis d'identifier un large spectre de raisons possibles allant de l'envie d'apprendre [Lakhani & Von Hippel, 03] jusqu'à la valorisation de la réputation et du statut professionnel [Lerner & Triole, 00], en passant par le désir de contribuer à l'esprit de la communauté du logiciel libre [Von Hippel & Von Krogh, 03]. La plupart d'entre-elles ont effectué leur analyse empirique soit en étudiant un cas de développement de logiciel open source (ex : Linux, Apache, Firefox, etc.), soit en effectuant, sous forme d'enquête, l'analyse de la fréquence des motivations déclarées par les développeurs interrogés.

Pour autant, outre les motivations, la façon dont ces programmeurs se comportent – leur *niveau de participation*, leur *implication*, leur *attitude* – demeure une interrogation digne d'une attention particulière. Ces lacunes, qui sont autant révélatrices de la jeunesse du sujet que du manque de cadre théorique, justifient l'intérêt de recherches oeuvrant pour une modélisation de ce processus de conception de logiciels [Scacchi *et al.*, 06].

A cet effet, l'objectif de cet article est de comprendre comment les développeurs se comportent dans un projet open source, d'identifier les antécédents et les effets produits. La partie empirique permettant d'éprouver cette problématique a été effectuée à partir d'une enquête conduite auprès de 50 projets open source hébergés sur SourceForge.net, actuellement la plus grande plateforme web de

¹ <http://www-128.ibm.com/developerworks/opensource>

² <http://www.sunsource.net/>

³ <http://oss.oracle.com/>

⁴ <http://opensource.arc.nasa.gov/>

projets open source. Les résultats de l'enquête révèlent en particulier à montrer que les comportements « positifs » des développeurs sont expliqués davantage par les motivations sociales qu'individuelles.

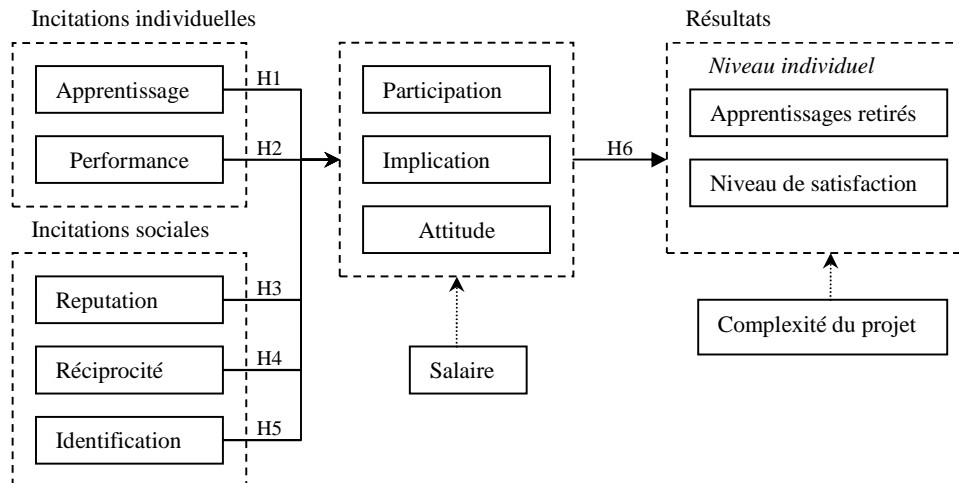
2 Analyse de la littérature

Les succès d'applications open source phares dans les systèmes d'exploitation (par exemples : Linux, Apache Server), des logiciels de bureautique (par exemples : Open Office, Firefox) ou encore dans les langages de programmation (par exemple : PHP-MySQL, Perl), peut donner l'impression que la plupart des projets de logiciels libres parviennent facilement à capter et maintenir la participation des contributeurs. Cependant, la réalité semble toute autre puisque beaucoup de projets sont abandonnés [Ewusi-Mensah, 97] car délaissés par leurs développeurs avant même que la version bêta de l'application ne soit publiée [Stewart & Gosain, 06].

Les projets open source sont ainsi exposés, d'une manière générale, au risque d'un *turn over* élevé [Von Hippel & Von Krogh, 03] et on estime que le rapport entre projet actifs / projets statiques peut être interprété dans les mêmes proportions que celles d'une loi de Pareto [Hunt & Johnson, 02]. Ce faisant, selon le comportement des développeurs, les résultats des projets open source peuvent être tout à fait différents et leur achèvement incertain.

Cette particularité du processus de développement des logiciels libres, par rapport au marché traditionnel, appelle les chercheurs en Systèmes d'Information à étudier conjointement les motifs et les comportements des développeurs dans les projets open source (voir modèle de recherche ci-après).

Figure 1 : Modèle de recherche



La plupart des recherches les plus célèbres menées sur les comportements des individus associés aux technologies de l'information [Davis, 89 ; Hartwick & Barki, 94 ; Venkatesh & Davis, 00, Venkatesh *et al.*, 03 ; Wixom & Todd, 05] ont construit leurs modélisations théoriques à partir des concepts et considérations de théories comme la *Theory of Reasoned Action* [Ajzen & Fishbein, 80], la *Theory of Planned Behavior* [Ajzen, 91; Taylor and Todd, 95] ou encore la *Social Cognitive Theory* [Compeau and Higgins, 95]. Ces travaux ont permis de mettre en lumière et d'éprouver les facteurs associés à des dimensions comme la satisfaction ou l'acceptation [Wixom & Todd, 05]. Ce faisant, plutôt que de chercher à construire de nouveaux concepts théoriques au nom d'une prétendue spécificité de notre objet de recherche (dont la justification aurait au demeurant toujours été discutable), il nous a semblé préférable d'asseoir notre travail sur des bases conceptuelles déjà largement admises par la communauté scientifique.

Ainsi nous mobiliserons les notions d'*attitude*, de *participation* et d'*implication* et nous considèrerons que plus un développeur sera motivé, plus son comportement (mesuré selon ces trois items) sera supposé actif et plus son résultat individuel (mesuré en termes d'apprentissage retiré et de niveau de satisfaction) sera attendu comme positif.

2.1 Le comportement

L'*attitude* correspond au jugement positif ou négatif émis par une personne face à un stimulus [Ajzen & Fishbein, 80]. Selon les auteurs, l'attitude va déterminer la disposition qu'aura une personne à accomplir la tâche en question ou bien à se comporter comme souhaité.

L'implication, quant à elle, se comprend comme "(...) a subjective psychological state, reflecting the importance and personal relevance of an object or event" [Barki & Hartwick, 89, p.61]. En d'autres termes, alors que l'attitude correspond à une donne affective d'un individu vis-à-vis d'un évènement ou d'un objet, l'implication est le reflet de l'importance perçue de ce dernier.

La participation, enfin, a été définie comme le fait d'un individu de prendre part ou de contribuer à quelque chose [Vrom & Jago, 98]. Dans le domaine de la conception de systèmes d'information, la participation fait référence à l'ensemble des activités que les concepteurs réalisent tout au long du processus de création de l'application.

A la différence de l'approche traditionnelle, la conception d'applications open source s'écarte de la démarche classique de type « top-down » puisque, les codes sources étant librement accessibles, n'importe quel utilisateur peut contribuer à l'améliorer ou encore à corriger des bugs sans pouvoir prétendre pour autant à une rétribution financière. Dans ce sens, les participants de projets open source se rapprochent d'un système communautaire [Von Hippel & Von Krogh, 03] sans relations autoritaires basées sur des statuts hiérarchiques. Ainsi, dans des cas comme celui du serveur Apache, des études, comme celle de Lakhani & Von Hippel (2003, p. 925), ont pu focaliser leur analyse sur les utilisateurs, ceux-ci étant en même temps les développeurs de l'application. Pour autant ces deux rôles ne sont pas systématiquement confondus et les résultats observés par les auteurs ne peuvent donc pas être généralisés à tous les projets open source. Gacek & Arief (2004) par exemple, montrent que durant le processus de développement plusieurs changements peuvent apparaître et qu'un utilisateur peut passer du stade passif, actif puis devenir développeur (ou inversement). Cette évolution des rôles a également été confirmée dans d'autres travaux comme ceux de Ye *et al.* (2004), ou encore Crowston *et al.* (2006). Malgré tout, l'objectif de notre article n'est pas d'étudier le caractère évolutif du rôle des participants d'un projet open source mais d'analyser à un moment donné du projet le lien entre motivations, comportements des développeurs et résultats. Nous formulerons donc l'hypothèse que les développeurs ayant un haut niveau de motivation par rapport à un projet, se comportent de manière active à son égard et contribuent de manière significative à ces différentes avancées.

2.2 Les variables indépendantes

Les recherches réalisées pour expliquer les motivations des développeurs à participer à des projets open source mobilisent principalement deux courants théoriques distincts. Le premier se concentre sur les facteurs sociaux de la *social exchange theory and social movement research* [Blau, 64] alors que le second se focalise davantage sur les aspects individuels et plonge ses racines dans le domaine de la psychologie [Dweck & Leggett, 88]. Faire la distinction entre les

motivations individuelles, d'une part, et sociales d'autre part, renvoie à la dialectique de Von Hippel & Von Krogh (2003) selon lesquels les communautés open source doivent être entendues comme des systèmes intermédiaires entre le modèle privé et le modèle collectif.

2.2.1 Les motivations individuelles

Notre recherche suggère que les motivations individuelles des développeurs peuvent être articulées autour de deux variables principales : les apprentissages espérés et la performance escomptée.

En effet, ces variables sont associées à la notion de *goal-oriented* initialement développée dans le domaine de la psychologie et des sciences de l'éducation (e.g., Diener & Dweck, 1978, 1980; Dweck, 1975; Dweck & Reppucci, 1973) pour expliquer les différences de comportement d'apprentissage au niveau des étudiants. « L'orientation objectif » de l'individu est devenue depuis une des variables les plus utilisées en psychologie et reste l'approche dominante pour étudier les motivations [Elliot & Dweck, 88]. Selon la théorie originelle (telle que définie par Dweck & Elliot, 83 ; Dweck & Leggett, 88) ce concept est divisible en deux catégories. La première correspond aux objectifs d'apprentissage que l'individu espère atteindre et aux nouvelles connaissances qu'il espère retirer. La seconde fait référence aux objectifs de performance visés par l'individu qui sont alors concentrés sur la vérification et l'évaluation de ses propres capacités à accomplir certaines tâches.

Les objectifs en termes d'apprentissage sont de loin considérés comme les premiers facteurs motivationnels que peuvent avoir les développeurs à rejoindre un projet open source. En effet, celui-ci représente un contexte pertinent au partage de connaissances, d'expertises comme d'astuces techniques [Raymond, 1999; Ljungberg, 2000; Kogut & Metiu, 2001; Lakhani *et al.*, 2003; Von Krogh, 2003; Stewart & Gosain, 2006]. Ce faisant, un développeur pourra valoriser ses compétences en termes de programmation, de maîtrise de langages de développement [Crowston *et al.*, 06] ou des règles professionnelles (par exemple : la façon dont les scripts doivent être structurés, dans quels contextes les fonctions et procédures doivent être appelées, les pratiques en termes de nommage des variables, etc.). Cet apprentissage correspond donc au *learning by doing* au sens de Brown and Duguid (1991) qui considèrent l'apprentissage comme un processus par lequel les apprenants sont également des pratiquants ; d'où l'hypothèse de recherche suivante :

Hypothèse 1 : *Les motivations d'apprentissage des développeurs influencent positivement leur comportement vis-à-vis du projet open source.*

Outre les apprentissages espérés, Dweck (1990) relevait que le comportement d'une personne est également façonné par son objectif de performance fixé. Dans

le domaine de l'acceptation des Technologies de l'Information, Venkatesh *et al.* (2003, p. 449) définissent ce dernier comme le degré auquel un individu croit qu'utiliser un système l'aidera à améliorer son travail personnel. Dans le cas des projets open source, les participants peuvent allouer plusieurs objectifs de performance à leurs contributions. Par exemple, Hann *et al.* (2004) ont observé auprès des 122 développeurs du système Apache que l'impact potentiel du projet sur la carrière professionnelle de ces programmeurs apparaissait comme un facteur déterminant de leur niveau de participation. Comme le font des développeurs en « free lance », les participants d'un projet open source peuvent rechercher une expérience professionnelle leur permettant ensuite de l'afficher comme une preuve de leurs compétences. Ceci nous autorise donc à formuler l'hypothèse suivante :

Hypothèse 2 : *Les objectifs de performance des développeurs influencent positivement leur comportement vis-à-vis du projet open source.*

2.2.2 Les motivations sociales

Une première forme d'incitation sociale pour les participants à un projet open source est relative au sentiment d'appartenance à la communauté. Ce besoin *d'identité* indique l'affiliation d'une personne à un groupe social. Comme Tajfel (1972) l'argumente : "*social identity is the individual's knowledge that he belongs to a certain social group together with some emotional or value significance to him of this group membership*" (Tajfel, 1972, p.292, In Hogg and Terry, 2000, p.122). Dans cette définition, il ressort deux éléments gagnant à être distingués : tout d'abord le niveau auquel l'individu se considère appartenir au groupe, puis le sentiment de fierté à faire ainsi partie du groupe social.

Dans les communautés open source ces sentiments représentent un élément fédérateur du travail des développeurs voire même un facteur explicatif de leurs comportements [Stewart & Gosaim, 06]. Cette influence sociale - essentielle au développement de projets devant ainsi être accomplis en dépit d'une autorité hiérarchique centrale et d'enjeux commerciaux [Scacchi *et al.*, 06] - a été analysée dans de nombreux articles [Gallivan, 2001 ; Von Krogh, 2003 ; Gosain, 2003 ; Crowston *et al.*, 2006].

En introduisant les notions de *normes*, de *valeurs* et de *croyances*, Stewart & Gosaim (2006) ont mis en lumière le fait qu'en partageant les principes idéologiques du logiciel libre (*vs* les logiciels propriétaires) les développeurs façonnent leurs comportements envers les projets open source (par exemple : la qualité du code produit, la rapidité avec laquelle les bugs vont être corrigés, la façon de réutiliser le code d'un autre développeur, etc.).

Hypothèse 3 : *Le sentiment d'identité à la communauté open source influence positivement le comportement des développeurs.*

Au-delà des principes idéologiques à partir desquels les communautés open

sources ont été fondées, le comportement des développeurs est également influencé par un autre facteur social : la *réciprocité*. Cette dernière correspond au fait qu'un individu contribue pour la communauté en espérant que les autres membres se comporteront de la même manière en retour [Shumaker & Brownell, 1984].

La réciprocité est une condition nécessaire à la stabilité d'une coopération et au maintien d'une confiance partagée (Arrow, 1976). Toutefois, à la différence de la théorie des jeux, au sein d'une communauté open source, celle-ci ne signifie pas des échanges « un à un », mais couvre un spectre plus collectif : « comportez vous envers les autres membres de la communauté, comme je l'ai fait avec vous. » Cependant, les réseaux électroniques par lesquels une communauté open source existe, font que cette dernière contraste avec les communautés de pratiques traditionnelles [Wasko & Faraj, 05, p. 37] où les relations de face-à-face fréquentes entre les individus renforcent les sanctions sociales si la réciprocité des relations n'est pas suffisamment assurée. Les communautés open source peuvent davantage être considérées comme des « réseaux de pratiques » [Brown & Duguid, 00] dans lesquels les échanges entre les membres sont médiatisés par un élément tiers (ici une plateforme web). Le capital social d'une communauté [Lin, 2001; Putnam, 1995] ne peut donc pas se retrouver intégralement sur un réseau électronique de pratiques [Nahapiet and Ghoshal, 1998]. Par exemple, dans une communauté open source, les participants n'ont pas de contrôle sur les forums de discussion et aucune assurance que la personne qu'ils aident ne se comportera pas comme une « cavalier solitaire » en n'assurant aucune réciprocité envers la communauté. D'ailleurs ces communautés électroniques de pratiques disparaîtraient si trop de comportements « opportunistes » (au sens de la théorie des jeux) se développaient [Wasko and Faraj, 05]. Ceci renforce l'intérêt d'une réciprocité satisfaisante par laquelle le comportement des développeurs open source peut se façonner.

Hypothèse 4 : La réciprocité influence positivement le comportement des développeurs dans les projets open source.

La *social exchange theory* [Blau, 1964] explique par ailleurs que les acteurs engagés dans des relations sociales espèrent également d'autres rétributions comme la qualité de la réputation acquise au sein du groupe et la reconnaissance par ses pairs. Dans les projets open source, construire sa réputation signifie obtenir le respect des autres membres de la communauté [Nichols & Twidale, 02]. La réputation des individus influence la confiance et représente donc un pouvoir de dissuasion des personnes impliquées dans la relation [Shapiro *et al.*, 1992]. Dans le cas de l'open source, l'influence de cette variable doit être davantage analysée du fait de la diversité des participants. Si la plupart des membres des équipes de projets sont plus souvent de simples utilisateurs que de véritables participants

actifs [Von Hippel & Von Krogh, 03], alors on peut supposer que la valorisation des réputations des membres ne concerneront que peu d'entre eux : les membres actifs. Les développeurs faisant justement partie de cette catégorie, l'hypothèse suivante est justifiée.

Hypothèse 5 : *La valorisation de la réputation escomptée au travers du projet open source influence positivement le comportement des développeurs.*

2.3 Variables dépendantes

Le succès des technologies de l'information reste de loin la variable dépendante la plus utilisée dans les recherches en systèmes d'information [Crowston *et al.*, 06]. Au niveau des travaux menés sur les logiciels open source, la même pratique semble être observée puisque leur « popularité » [Stewart & Ammeter, 02] a été d'abord mesurée à l'aide d'indicateurs comme le nombre d'utilisateurs, le nombre de téléchargements effectués de l'application, ou encore le nombre de pages consultées du site. Pour autant, les spécificités du développement en open source restent entachées d'ambiguïtés quant à la pertinence de ce genre de mesures. Les travaux conduits par Crowston *et al.* (2006) ont été, à ce titre, révélateurs et ont montré que l'on ne pouvait pas utiliser des indices de performance classiques comme ceux employés pour l'évaluation du succès des logiciels d'éditeurs. On retiendra par exemple que le code source des applications ainsi développées étant libre, il est fréquent que des modules de programmes efficaces soient par la suite réutilisés dans d'autres projets. Si bien que c'est davantage dans la diffusion du tout ou partie du code dans d'autres projets qu'il conviendrait dès lors d'analyser la popularité d'une application open source. La difficulté d'un tel « traçage » de lignes de programmes, au demeurant librement modifiables par tout internaute, invite alors à se détacher d'une mesure de la performance centrée sur le logiciel effectivement produit.

Ainsi, d'autres travaux ont davantage cherché à éprouver la performance des projets open source sous un angle plus processuel. Crowston & Scozzi (2002) se sont par exemple intéressés aux versions (alpha, beta, etc.) au travers desquelles l'application a été déclinée dans le temps. Des variables focalisées sur le processus de développement (niveau d'activité, productivité au niveau des lignes de code produites, rapidité de correction de bugs, etc.) sont depuis mises en avant [Crowston *et al.* 06]. Dans cette tendance, l'analyse du succès d'un projet open source est donc centrée sur le travail du développeur [Scacchi, 02] dont la satisfaction des motivations reste la pierre angulaire [Crowston, 06]. Pour autant, force est de constater, que la majorité des travaux a plus traité cette variable comme un *input* (considérés comme un état factuel) que comme un *output* dont les déterminants restaient à découvrir. Ce point est d'autant plus important que les « bénéfiques » retirés par les participants constitueront, par la suite, des éléments structurant de leur implication dans d'autres projets et donc, d'une manière

générale, de la pérennité de la communauté open source.

Dans notre modèle de recherche, les variables dépendantes ont été construites autour de deux éléments que nous avons considérés ici comme pivots : les apprentissages effectivement retirés par le développeur et son niveau de satisfaction global envers le projet.

2.3.1 Les apprentissages retirés

Comme nous l'avons introduit auparavant, les apprentissages escomptés par un individu façonnent la manière avec laquelle celui-ci va se comporter (et ce, en termes de *participation*, d'*implication* et d'*attitude*) [Dweck & Leggett, 88]. Les changements induits dans son travail quotidien, les connaissances qu'il va mobiliser et les compétences qu'il va développer influencent en retour les apprentissages effectivement retirés de son expérience.

Les résultats récents révélés par Hemetsberger & Reinhardt (2006) au niveau des projets open source, tendent à montrer que le transfert, tout comme la création de connaissances, sont possibles du fait de la mise en partage du code produit et du stockage des échanges réalisés entre les participants pour le construire. Ces auteurs expliquent que certaines fonctionnalités offertes par les plateformes hôtes compensent, dans une certaine mesure, l'absence de relations en face-à-face entre les membres du projet. Par exemple, les systèmes de *versioning* CVS ou encore les commentaires ajoutés dans les programmes, permettent aux développeurs d'avoir accès à la trame du processus de développement ayant abouti au code produit. Cet historique permet alors de capitaliser sur l'expérience des autres participants en termes d'erreurs et d'astuces de programmation.

Pour distinguer les niveaux d'apprentissages recueillis par les intéressés, nous nous sommes appuyés sur les travaux de Hogg & Terry (2000) puis de Kankanhalli (2005) qui permettent de nuancer sur trois niveaux les processus de transfert et de création de connaissances. La première, la *réplication*, s'entend comme le fait qu'une personne maîtrise de manière plus efficiente les techniques et méthodes relatives à la réalisation de ces tâches courantes. La deuxième, *l'adaptation*, est relative à la capacité de l'individu à actualiser ses connaissances de manière à adapter son mode résolutoire à des changements de l'environnement. La dernière, *l'innovation*, correspond à l'aptitude de l'individu à transformer ses schémas de pensée, à créer de nouveaux modes résolutoires (et donc de nouvelles connaissances) représentant une valeur ajoutée originale et significative. On retrouve donc en filigrane de ce triptyque les notions d'*exploitation* et d'*exploration* [March, 91] relatives à l'apprentissage individuel sans toutefois le caractère dual dans lequel les deux ont souvent été réduites.

Hypothèse 6 : *Les développeurs ayant un comportement positif envers un projet open source reconnaissent recueillir des capacités de réplication.*

Hypothèse 7 : *Les développeurs ayant un comportement positif envers un projet open source reconnaissent recueillir des capacités d'adaptation*

Hypothèse 8 : *Les développeurs ayant un comportement positif envers un projet open source reconnaissent recueillir des capacités d'innovation*

2.3.2 Le niveau de satisfaction

Comme nous l'avons développé précédemment, les motivations des participants à un projet open source couvrent une dimension sociale aussi bien qu'individuelle. Ce faisant, un développeur pourra avoir des éléments de satisfaction de par son expérience sociale vécue avec l'équipe de développeurs participant au projet.

Hypothèse 9 : *Les développeurs ayant un comportement positif envers un projet open source affichent une satisfaction vis-à-vis de l'équipe*

De même, un développeur pourra avoir une satisfaction plus ou moins importante envers l'aboutissement effectif du projet. Cet élément est d'autant plus important à prendre en considération que la plupart des projets open source sont abandonnés avant leur terme [Stewart & Gosain, 06] ou bien que les résultats atteints ne correspondent pas aux objectifs tels qu'ils avaient été définis initialement [Scacchi, 02]. Sur ce dernier point, les développeurs sont d'ailleurs bel et bien les investigateurs de ces sortes de « dérives téléologiques » et, chemin faisant, les développements réalisés peuvent dessiner un fil directeur différent de celui par lequel le projet avait été initié [Roberts *et al.*, 04]. Les relations non hiérarchiques à l'intérieur de l'équipe projet conduisent donc souvent à un phénomène d'appropriation par les développeurs des objectifs pouvant ou devant être atteints par le projet.

Hypothèse 10 : *Les développeurs ayant un comportement envers un projet open source affichent une satisfaction vis-à-vis du projet réalisé*

2.4 Variables de contrôle

Etant donné que les incitations à participer à un projet open source peuvent varier considérablement selon les individus et le contexte du projet, nous avons inclus deux variables de contrôle dans notre modèle : la complexité du projet et les contributions financières éventuellement reçues.

Les enjeux perçus quant aux objectifs associés au logiciel à réaliser peuvent être attendus comme éléments structurant de la participation et du comportement des développeurs. Par exemple, le développement d'un package ERP représente, *a priori*, un niveau de sophistication plus important qu'un simple patch de correction d'un logiciel de bureautique. Ce faisant, selon le niveau de complexité de l'application à réaliser, des différences peuvent être attendues sur des éléments comme la durée du projet, la composition de l'équipe de développeurs, leurs motivations et donc leurs comportements.

D'autre part, il a été montré [Lakhani *et al.*, 02] que bien que la majorité des contributeurs des communautés open source sont des développeurs professionnels ou amateurs agissant à titre individuel, certains sont employés d'organisations finançant leur participation. Cette dérogation au mode de développement du logiciel libre est confirmée par des études ayant montré que, dans certains projets, plus de la moitié des contributeurs étaient en partie rémunérés pour leur travail [Lakhani *et al.* 2002, Hertel *et al.* 2003]. Dès lors il convient de retenir l'éventualité d'une rétribution financière comme élément perturbateur de l'influence des variables préalablement décrites sur le comportement des développeurs.

3 Design de la recherche

La plupart des recherches conduites sur les projets open source ont abondé la connaissance scientifique de célèbres cas de succès comme Linux ou Apache [Gallivan 2001; Mockus *et al.* 2002, Bagozzi *et al.*, 2006]. Même si de tels succès ainsi relatés ont contribué à une mise en lumière des spécificités des communautés du logiciel libre, ils n'en demeurent pas moins non représentatifs de la majorité des projets [Stewart & Gosaim. 2006] ne serait-ce qu'en termes de technologie, d'enjeu, de nombre de participants, etc.

Notre dispositif de recherche nous a permis de collecter les données des projets hébergés sur la plateforme SourceForge.net dont le fait d'être le premier site d'accueil de communautés open source (en juillet 2005 plus de 124 900 projets et 1,3 millions d'utilisateurs étaient répertoriés) suffit à justifier le choix ainsi fait. Ce succès du site peut probablement être imputé, d'une part, au fait que (fidèlement à la philosophie open source) le site héberge gratuitement les projets de logiciels libres et que, d'autre part, il offre aux membres une infrastructure leur permettant de contrôler et manager les étapes du développement (gestion du versioning, outils de communication et de partage de code, etc.) En d'autres termes, SourceForge fournit à tous les projets un package de fonctionnalités standards réduisant du même coup dans notre enquête les variances de comportement des développeurs pouvant être dues aux différences de technologies mises à disposition des développeurs d'un projet à un autre.

Les développeurs de SourceForge ont été sollicités par le biais de deux dispositifs d'enquête. Le premier dressait une liste de questions ouvertes aux administrateurs de projet quant à leur façon de développer ainsi gratuitement des logiciels. 100 administrateurs ont été de la sorte contactés par e-mail, 34 d'entre eux ont répondu au questionnaire. Les données qualitatives ainsi recueillies ont été utilisées afin d'inférer les facteurs de motivation gagnant effectivement à être intégrés dans le second dispositif d'enquête.

Nous avons alors sélectionné les projets cibles dans une seule et même catégorie

(« code generator, design and framework ») limitée aux applications destinées aux entreprises. Ces mesures restrictives ont permis, ce faisant, de contrôler les variances pouvant apparaître du fait des natures différentes des projets. Nous avons également vérifié que les projets ciblés étaient à proprement parler actifs (et non pas en sommeil) de par les morceaux de codes récemment postés, les dernières mises à jour et corrections de bugs faites, les échanges sur le forum de discussion, etc.

Au total, 50 projets ont ainsi été retenus au sein desquels nous avons pu solliciter les membres identifiés comme développeurs. 92 sur les 310 contactés ont répondu à l'enquête (soit un taux de réponse de 29,7%).

La plupart de nos construits utilisés pour évaluer les variables relatives aux motivations - aux comportements et à la performance des individus, ont été établis en référence à une analyse des instruments de mesure déjà utilisés dans des articles de référence (en particulier : Hartwick & Barki, 94 ; Zhao & Deek, 2004 ; Gray & Meinster 2004 ; Davis, 2003). A ce titre, dans le questionnaire, tous les items ont été mesurés avec une échelle de Likert à 5 valeurs (1 : « pas du tout d'accord » – 5 : « tout à fait d'accord »).

4 Résultats

Afin de fiabiliser la validité des relations établies par notre traitement de données nous avons procédé aux deux étapes analytiques de l'analyse PLS [Hair *et al.* 1998] : (1) le modèle de mesures qui inclut la fiabilité et la validité des discriminants ; puis (2) le modèle structurel. Sur ce dernier, une analyse factorielle exploratoire (EFA) a d'abord été faite pour vérifier que les facteurs structurels proposés étaient compatibles avec les données collectées (voir tableau 1). Ces dernières ont ensuite été traitées avec une analyse en composantes principales (avec rotation selon la méthode Varimax et une normalisation Kaiser).

Enfin, une analyse factorielle confirmatoire (CFA) a pu être réalisée avec SPSS pour tester le modèle. Les propriétés des mesures effectuées sont résumées dans le tableau 2.

Tableau 1: Résumé des construits

Construct Name	Construct identifier	Cronbach Alpha	Initial number of items	Number of items carried forward the analysis	AVE	CR
Learning outcomes	LEA	0,89	3	3	0,82	0,94

Performance expectancy	PER	0,72	4	3	0,73	0,88
Reputation	REP	0,75	4	3	0,76	0,84
Reciprocity	REC	0,88	4	4	0,73	0,92
Identity	IDE	0,73	4	3	0,72	0,82
Participation	PAR	0,75	8	5	0,67	0,88
Involvement	INV	0,84	9	6	0,68	0,89
Attitude	ATT	0,87	4	4	0,73	0,92
Adaptation	ADA	0,81	2	2	0,84	0,91
Replication	REP	0,90	2	2	0,91	0,95
Innovation	INN	0,72	2	2	0,79	0,88
Team satisfaction	TES	0,87	2	2	0,88	0,94
Project satisfaction	PRS	0,83	3	3	0,78	0,91
Project complexity	PRO	0,74	4	3	0,72	0,87

Tableau 2 : Tableau des mesures

Latent Construct	Indicators	Standard Loading	Weight	Latent Construct	Indicators	Standard Loading	Weight	
LEA	LEA 1	0,911	0,389	INV	INV 3	0,747	0,236	
	LEA 2	0,868	0,296		INV 4	0,722	0,160	
	LEA 3	0,948	0,409		INV 5	0,816	0,238	
PER	PER 2	0,834	0,531		INV 6	0,825	0,234	
	PER 3	0,794	0,391		INV 7	0,835	0,195	
	PER 4	0,758	0,324		INV 8	0,798	0,246	
REP	REP 1	0,822	0,453		ATT	ATT 1	0,871	0,262
	REP 2	0,826	0,487			ATT 2	0,849	0,292
	REP 3	0,787	0,285	ATT 3		0,864	0,236	
REC	REC 1	0,854	0,284	ATT 4		0,831	0,383	
	REC 2	0,812	0,205	ADA	ADA 1	0,928	0,578	
	REC 3	0,902	0,305		ADA 2	0,906	0,510	
IDE	IDE 1	0,791	0,396	REP	REP 1	0,947	0,495	
	IDE 2	0,789	0,477	INN	REP 2	0,958	0,555	
	IDE 4	0,777	0,398		INN1	0,922	0,642	
PAR	PAR 1	0,646	0,176	TES	INN2	0,854	0,476	
	PAR 3	0,727	0,249		TES1	0,949	0,562	
	PAR 4	0,752	0,259	TES2	0,934	0,498		
	PAR 5	0,779	0,348	PRS	PRS1	0,899	0,384	
	PAR 8	0,740	0,412		PRS2	0,905	0,333	
			PRS3		0,834	0,422		

Les items apparaissent avec un Alpha de Crombach supérieur à 0.72 et sont donc

considérés comme adéquats [Fornell & Larcker, 81]. Chacun des poids étant, d'autre part, statistiquement significatif ($p < 0.001$), l'uniformité de leur distribution montre l'égalité de la contribution de chacun des items au construit.

Une confirmation de la fiabilité est ici donnée par la *composite reliability* (CR) dans l'analyse factorielle confirmatoire. Cette mesure, fréquemment utilisée pour tester la validité d'un modèle [Raghunathan *et al.*, 1999], et indique dans notre cas une forte validité interne (voir tableau 2).

Les résultats fournis par l'analyse PLS (voir tableau 3 et figure 2) ont été épurés des relations non significatives de manière à ne pas obscurcir la lisibilité des résultats. Comme cela est ainsi retranscrit sur la figure 2, les motivations individuelles (apprentissage escompté, performance attendue) et sociales (identité, réciprocité, réputation) ont des effets significatifs sur le comportement des développeurs (participation, implication, attitude) avec respectivement des coefficients de 0,100, 0,119, 0,468, 0,228, 0,136. L'ensemble de ces variables explique à lui seul 49% de la variance du comportement des développeurs.

Les motivations individuelles et sociales couplées avec le comportement du développeur explique les bénéfices retirés (apprentissage effectif : adaptation, innovation, réplication ; satisfaction : envers l'équipe, envers le projet) à hauteur de 16,4% 20,5%, 17,4%, 11,9%, 11,2%. Pour leur part, les variables de contrôle exercent une influence de 13,3% et 11,3%.

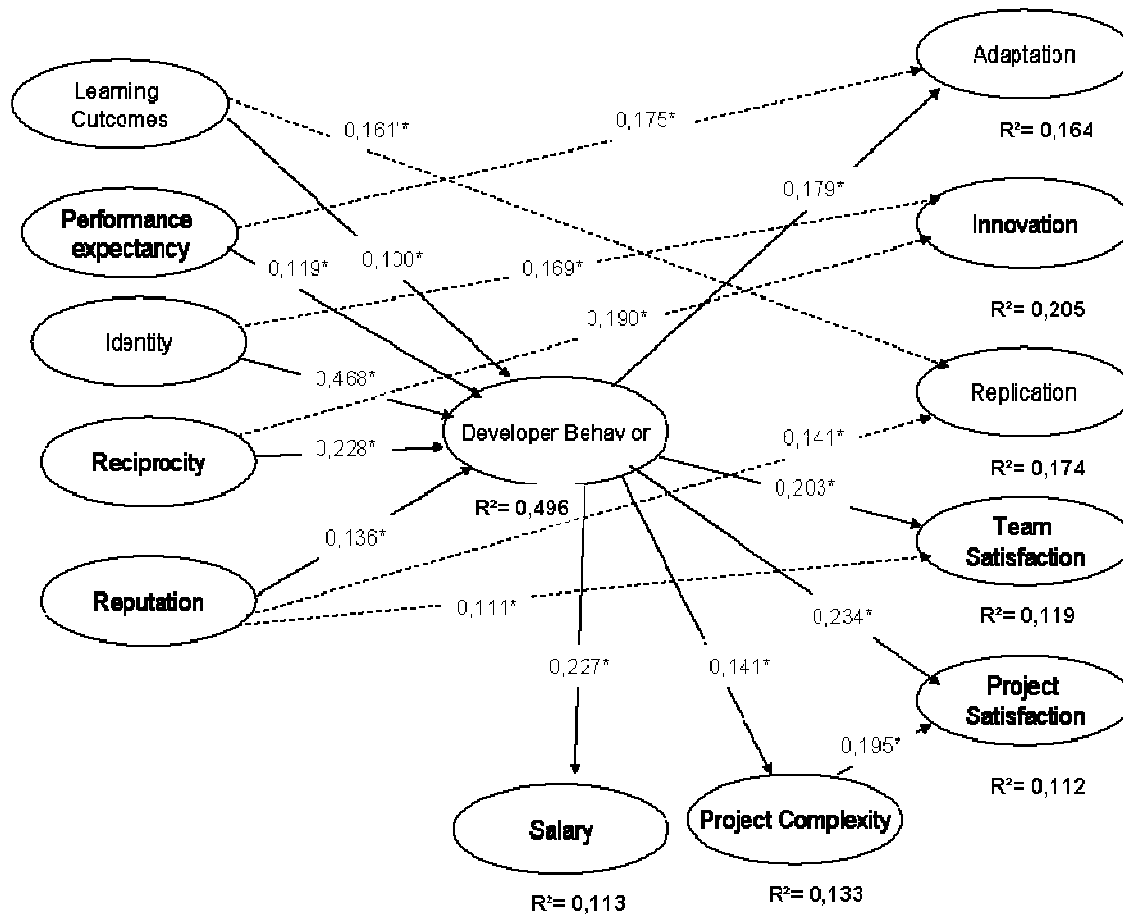
Tableau 3 : Corrélation des variables latentes

	Learning outcomes	Performance expectancy	Reputation	Reciprocity	Identity	Salary	Replication	Adaptation	Innovation	Project Satisfaction	Team satisfaction	Project complexity
Learning outcomes	0,905											
Performance expectancy	0,53*	0,854										
Reputation	0,191*	0,661*	0,871									
Reciprocity	0,206*	0,529*	0,536*	0,854								
Identity	0,523*	0,169*	0,197*	0,261	*	0,848						
Salary	0,229*	0,004*	0,142*	0,150	0,123	*	0,818					
Replication	0,224*	0,235*	0,264*	0,247	0,198	0,295	*	0,824				
Adaptation	0,159*	0,296*	0,275*	0,243	0,139	0,091	0,678	*	0,854			

Innovation	0.120*	0.258*	0.249*	0.353	0.216	0.070	0.527	0.452	0.916		
Project Satisfaction	0.094*	0.153*	0.149*	0.200	0.136	0.127	0.358	0.347	0.374	0.953	
Team satisfaction	0.138*	0.153*	0.180*	0.116	0.060	0.034	0.316	0.283	0.252	0.516	0.888
Project complexity	0.168*	0.157*	0.262*	0.323	0.041	0.055	0.306	0.203	0.332	0.239	0.156
				*	*	*	*	*	*	*	*
											0.938

The numbers between parentheses are the square root of AVE, and * p<0,001

Figure 2 : Modèle structurel



5 Discussion et conclusion

Cette recherche présente les limites inhérentes à la méthodologie analytique utilisée ainsi qu'à la démarche hypothético-déductive suivie et qui constituent autant de pistes pour l'utilisation d'autres dispositifs d'enquête pour éprouver le modèle de recherche conçu. On pourra en particulier regretter que les variables dépendantes de notre modèle ne soient davantage expliquées (le comportement du développeur explique 16% des apprentissages retirés par le développeur et un peu moins de 12% de sa satisfaction globale).

Toutefois, nos résultats offrent un éclairage supplémentaire sur le rôle des facteurs de motivation individuels (apprentissage escompté, performance attendue) et sociaux (réputation, réciprocité, identification) sur le comportement des développeurs et des résultats associés en termes de performance ressentie au niveau du projet (apprentissage effectif, niveau de satisfaction). En effet, l'analyse empirique soutient nos hypothèses et révèle que les facteurs de motivation retenus dans notre modèle de recherche expliquent de manière statistiquement significative à près de 50% la variance du comportement des développeurs ($R^2=0,496$).

Cette étude peut donc être considérée comme une contribution supplémentaire à l'éclairage du « phénomène open source » et des facteurs de motivation des développeurs. Le comportement des développeurs apparaît ici comme influencé conjointement par les motivations individuelles et par les motivations sociales. Cependant, nos résultats montrent que les secondes représentent un facteur explicatif plus élevé que les premières. A ce titre, l'*identification* apparaît comme la variable la plus influente (0,468) ce qui confirme les observations faites par Stewart & Gosain (2006) sur la dimension idéologique de la communauté open source. C'est ainsi le besoin d'appartenir à un groupe partageant les valeurs du logiciel libre qui semble être le facteur de motivation le plus déterminant du comportement des développeurs.

Ce constat invite donc à relativiser certains « lieux communs » quant aux objectifs permettant d'expliquer que des développeurs acceptent de travailler bénévolement à la création de logiciels devant, qui plus est, ne pas faire l'objet d'une commercialisation. Les communautés open source sont, en effet, souvent amalgamées à des espaces d'accueil de programmeurs *freeware* avides d'expériences permettant d'alimenter les lignes de leurs CV. Or, nos résultats tendent à montrer que ce genre d'expectatives individuelles reste au second plan par rapport à la quête identitaire des membres de la communauté open source. Cette dimension d'ordre psychologique ainsi mise en exergue représente une piste pour d'autres travaux plus approfondis sur le facteur idéologique associé au comportement des développeurs.

Références

- [Ajzen & Fishbein, 80] Ajzen, I., Fishbein, M. (1980), *Understanding attitudes and predicting social behavior*, Englewood Cliffs, NJ: Prentice-Hall Inc.
- [Ajzen, 91] Ajzen, I. (91), "The theory of planned behavior", *Organizational Behavior and Human Decision Processes*, vol. 50
- [Arrow, 74] Arrow, K. J. (74), *The Limits of Organization*, W. W. Norton & Company, New-York
- [Bagozzi & Uptal, 06] Bagozzi, R. P., Utpal, M. D. (06), "Open Source Software User Communities: A Study of Participation in Linux User Groups", *Management Science*, vol. 52, n°7, pp. 1099-1115
- [Barki & Hartwick, 89] Barki, H., Hartwick, J. (89) « Rethinking the concept of user involvement », *MIS Quarterly*, vol. 13, n°1
- [Blau, 64] Blau, P. M. (64), *Exchange and Power in Social Life*, Wiley, New York
- [Bonaccorsi & Rossi, 03] Bonaccorsi, A., Rossi, C. (03), "Why Open Source Software Can Succeed", *Research Policy*, vol. 32
- [Brown & Duguid, 00] Brown, J. S., Duguid, P. (00), *The Social Life of Information*, Harvard Business School Press, Boston
- [Brown & Duguid, 91] Brown, J. S., Duguid, P. (91), "Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning, and Innovation", *Organization Science*, vol. 2, n°1
- [Compeau & Higgins, 95] Compeau, D. R., Higgins, C. A. (95), "Application of Social Cognitive Theory of Training Motivation: A Meta-Analytic Path Analysis of 20 Years of Training Research", *Journal of Applied Psychology*, vol. 85, n°5
- [Crowston & Scozzi, 02] Crowston, K., Scozzi, B. (02), "Open source software projects as virtual organizations: Competency rallying for software development", *IEE Proceedings Software*, vol. 149, n°1
- [Crowston et al., 06] Crowston, K., Howison, J., Annabi, H. (06) "Information systems success in free and open source software development: Theory and measures", *Software Process: Improvement and Practice* (Special Issue on Free/Open Source Software Processes.).
- [Davis, 89] Davis F. D. (89), "Perceived Usefulness, Perceived Ease of Use and User Acceptance of IT", *MIS Quarterly*, vol. 13, n°3
- [Diener & Dweck, 78] Diener, C.I., Dweck, C.S. (78), "An analysis of helplessness: Continuous changes in performance, strategy and

achievement cognitions following failure”, *Journal of Personality and Social Psychology*, vol. 36, pp. 451-462.

- [Diener & Dweck, 80] Diener, C.I., Dweck, C.S. (1980). An analysis of learned helplessness II. The processing of success. *Journal of Personality and Social Psychology*, 39, 940-952.
- [Diener & Elliott, 83] Dweck, C.S., Elliott, E.S. (83), “Achievement motivation”, in P. Mussen and M. Hetherington (dir.), *Handbook of child psychology* (Vol. IV, p.643-691). New York, NY: John Wiley.
- [Diener & Repucci, 73] Dweck, C.S., Repucci, N.D. (73), “Learned helplessness and reinforcement responsibility in children”, *Journal of Personality and Social Psychology*, vol. 25, pp. 109-116
- [Dweck & Leggett, 88] Dweck, C. S., Leggett, E. L. (88), “A social-cognitive approach to motivation and personality”, *Psychological Review*, vol. 95, n°2, pp. 256-273
- [Dweck & Leggett, 88] Dweck, C.S., Leggett, E.L. (88) , “A social-cognitive approach to motivation and personality”, *Psychological Review*, vol. 95, n°2, p. 256-273
- [Dweck, 75] Dweck, C.S. (75), “The role of expectations and attributions in the alleviation of learned helplessness”, *Journal of Personality and Social Psychology*, vol. 36, pp. 674-685
- [Fornell & Larcker, 81] Fornell C. & Larcker D. (1981) “ Evaluating structural equation models with unobservable variables and measurement error ”, *Journal of Marketing Research*, vol. 18, pp. 39-50
- [Gacek & Arief, 04] Gacek, C., Arief, B. (04), “The many meanings of open source”, *IEEE Software*, vol. 21, n°1
- [Gallivan, 01] Gallivan, M. J. (01), “Striking a balance between trust and control in a virtual organization”, *Information Systems Journal*, vol. 11
- [Gosain, 03] Gosain, S. (03), “Looking through a Window on Open Source Culture: Lessons for Community Infrastructure Design,” *Systèmes d’Information et Management*, vol. 8, n°1
- [Gray & Meister, 04] Gray, P.H., Meister, D.B. (04) “Knowledge Sourcing Effectiveness”, *Management Science*, vol. 50, no. 6, pp. 821-834
- [Hair et al., 98] Hair, J.F., Anderson, R.E., Tatham, R.L., Black, W.C. (98), *Multivariate Data Analysis with Readings*, Prentice Hall: Englewood Cliffs, NJ.

- [Hann *et al.*, 04] Hann, I.-H., Roberts, J., Slaughter, S., Fielding, R. (04), "Economic incentives for participating in open source software projects", *International Conference on Information Systems*
- [Hartwick & Barki, 94] Hartwick, J., Barki, H. (94), "Explaining the Role of User Participation in Information System Use", *Management Science*, vol. 40, n°4
- [Hemetsberger & Reinhardt, 06] Hemetsberger, A., Reinhardt, C. (06), "Learning and Knowledge-building in Communities", *Management Learning*, vol. 37, n°2
- [Hertel *et al.*, 03] Hertel, G., Neidner, S., Hermann, S. (03), "Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel", *Research Policy*, vol. 32, n°7
- [Hogg & Terry, 00] Hogg, M.A., Terry, D.J. (00). "Social Identity and Self-Categorization Processes in Organizational Contexts," *Academy of Management Review*, vol. 25, n°1
- [Hunt & Johnson, 02] Hunt, F., Johnson, P. (2002), "On the Pareto Distribution of Open Source Projects", *Proceedings of the Open Source Software Development Workshop*, Newcastle
- [Kankanhalli *et al.*, 05] Kankanhalli, A., Tan, B.C.Y., Kwok-Kee, W. (05) "Contributing Knowledge to Electronic Knowledge Repositories: an Empirical Investigation", *MIS Quarterly*, vol. 29, n°1, pp. 113-143.
- [Kogut & Metiu, 01] Kogut, B., Metiu, A. (01), "Open source software development and distributed innovation", *Oxford Rev. Econom. Policy*, vol. 17, n°2
- [Lakhani & Von Hippel, 03] Lakhani, K.; Von Hippel, E. (03) "How open source software works: free user-to-user assistance", *Research Policy*, vol. 32, n°6
- [Lakhani *et al.*, 02] Lakhani, K.R., Wolf, B., Bates, J., DiBona, C. 2002. The Boston Consulting Group, Hacker Survey, July, <http://www.bcg.com/opensource/BCGHackerSurveyOSCON24July02v073.pdf>
- [Lerner & Triole, 00] Lerner, J., Triole, J. (00), "The simple economics of Open source", *MIT Working Paper*, n°7600
- [Lin, 01] Lin, N. (01), *Social Capital*, Cambridge University Press
- [Ljungberg, 00] Ljungberg, J. (99), "Open source movements as a model for organising", *European Journal of Information Systems*, vol. 9
- [March, 91] March, J. G. (91), "Exploration and Exploitation in Organizational Learning", *Organization Science*, vol. 2, n°1.

- [Mockus *et al.*, 02] Mockus, A., Fielding, R., Herbsleb, J. D. (02), "Two Case Studies of Open Source Software Development: Apache and Mozilla", *ACM Transactions on Software Engineering and Methodology*, vol. 11, n°3
- [Nahapiet & Ghoshal, 98] Nahapiet, J., Ghoshal, S., (98), "Social Capital, Intellectual Capital, and the Organizational Advantage," *Academy of Management Review*, vol. 23, n°2
- [Nichols & Twidale, 02] Nichols, D., Twidale, M. "Usability and Open Source Software", *Working Paper 10/02*, Department of Computer Science - University of Waikato.
- [Putnam, 95] Putnam, R. (95), "Bowling Alone: America's Declining Social Capital," *Journal of Democracy*, vol. 6
- [Raymond, 99] Raymond, E. S. (99), *The Cathedral and the Bazaar. Musings on Linux and Open Source by an Accidental Revolutionary*, Sebastopol, CA: O'Reilly and Associates
- [Scacchi *et al.*, 06] Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S., and Lakhani, K. (06), "Understanding Free/Open Source Software Development Processes", *Software Process—Improvement and Practice*
- [Scacchi, 02] Scacchi, W. (02), "Understanding the requirements for developing Open Source Software systems", *IEE Proceedings Software*, vol. 149, n°1
- [Shapiro *et al.*, 92] Shapiro, D., Sheppard, B., Cheraskin, L. (92), "Business on a handshake", *Negotiation Journal*, vol. 8, n° 4
- [Shumaker & Brownell, 84] Shumaker, S., Brownell, A. (84), "Toward a Theory of Social Support: Closing Conceptual Gaps", *Journal of Social Issues*, vol. 40, n°4
- [Stewart & Ammeter, 02] Stewart, K. J., Ammeter, T. (02), "An exploratory study of factors influencing the level of vitality and popularity of open source projects", *Proceedings of the International Conference on Information Systems*.
- [Stewart & Gosain, 06] Stewart, K. J., Gosain, S. (06) "The Impact of Ideology on Effectiveness in Open Source Software Development Teams", *MIS Quarterly*, vol. 30, n°2.
- [Tajfel, 72] Tajfel, H. (72) "Experiments in a vacuum", in J. Israel, & H. Tajfel (Eds.), *The context of social psychology*. London: Academic Press
- [Taylor & Todd, 95] Taylor, S., Todd, P. A. (95), "Understanding Information Technology Usage: A Test of Competing Models", *Information Systems Research*, vol. 6 n°4

- [Venkatesh *et al.*, 03] Venkatesh V., Morris M. G., Davis G. B., Davis F. D. (03), "User acceptance of information technology: toward a unified view", *MIS Quarterly*, vol. 27, n°3
- [Von Hippel & Von Krogh, 03] Von Hippel, E., Von Krogh, G., (03) "Open Source Software and the Private-Collective Innovation Model", *Organisation Science*, vol. 14, n°2
- [Von Krogh, 03] Von Krogh, G. (03), "Open-Source Software Development", *MIT Sloan Management Review*, Spring
- [Vrom & Jago, 98] Vrom, D. J., Jago, A. G. (98), *The new leadership. Managing participation in organizations*, Inc, Prentice Hall
- [Wasko & Faraj, 05] Wasko, M., Faraj, S. (05), "Why Should I Share? Examining Social Capital and Knowledge Contribution in Electronic Networks of Practice", *MIS Quarterly*, vol. 29, n°1
- [Wixom & Todd, 05] Wixom, B. H., Todd, P. A. (05), "A Theoretical Integration of User Satisfaction and Technology Acceptance", *Information Systems Research*; vol. 16, n°1
- [Ye *et al.*, 04] Ye, Y., Nakajoki, K., Yamamoto, Y., Kishida, K., (04) "The Co-Evolution of Systems and Communities in Free and Open Source Software Development", in S. Koch (ed.), *Free/Open Source Software Development*, Idea Group Publishing, Hershey, PA
- [Zhao & Deek, 04] Zhao, L., Deek, F. (04), "User Collaboration in Open Source Software Development", *Electronic Markets*, vol. 14, n°2, pp. 89-103